



National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

THE UNIVERSITY OF ALBERTA

**Three-Dimensional Display of
Medical Images**

by



Yiu-Wing Tam

• A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Spring 1989

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-53018-9



THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Yiu-Wing Tam

TITLE OF THESIS: Three Dimensional Display of Medical Images

DEGREE FOR WHICH THIS THESIS WAS PRESENTED: Master of Science

YEAR THIS DEGREE GRANTED: 1989

Permission is hereby granted to The University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed)

Yiu - Wing Tam.

Permanent Address:

1235 Sandpiper Rd NW
Calgary, Alberta
Canada
T3K 2V7

Date: Spring 1989

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **Three Dimensional Display of Medical Images** submitted by Yiu-Wing Tam in partial fulfillments for the degree of **Master of Science**.

W. A. Davis

Dr. W. A. Davis
Supervisor

Xiaohu Li

Dr. X. Li

Barry Joe

Dr. B. Joe

W. Bischof

Dr. W. Bischof
External

H. Richard Hooper

Dr. R. Hooper
External

Date: October 1988

To

*God,
my parents, and
my wife Regina ...*

ABSTRACT

Three-dimensional display of medical data usually involves three steps: (1) image segmentation, (2) merging of 2D data into a 3D representation, and (3) display of the 3D object. This thesis gives a brief discussion on image segmentation based on edge detection and thresholding, and object representation and display techniques. Of the three steps mentioned, creating shaded graphics display of objects is computationally the most expensive task. This thesis proposes two methods, one method is aimed at producing images efficiently (the *voxel-based method*) and another method aimed at producing high quality shaded images (the *normal-averaging method*). Both methods are subsets of the gradient shading method. The gradient shading method uses both the distance of the object from the light source and the object surface orientation as the basis for shading. The theory and its implementation are discussed and images produced by several display techniques are presented and compared.

ACKNOWLEDGEMENTS

With sincere gratitude, I wish to thank my supervisor Dr. Wayne A. Davis for sharing with me his vast knowledge in Image Processing. I would like also to thank for his patience throughout the research. His guidance and encouragement have been invaluable.

I am also grateful to the members of my examining committee, Dr. X. Li, Dr. B. Joe, Dr. W. Bischof and Dr. R. Hooper, for their helpful comments.

I would like to thank Dr. R. Hooper and the Department of Nuclear Medicine, Cross Cancer Institute, Edmonton, Alberta for providing image data for experimentation throughout the research.

I wish also to thank the Advanced Technologies Department of the Alberta Research Council, Calgary, Alberta for providing the computing equipment for completing this thesis. Special thanks go to Dr. Brian Barge (Vice President, Alberta Research Council), Dr. Ernest Chang (Department Head, Advanced Technologies Department) and Mr. Maurice Engler (Pelorus Inc.) for their constant encouragement.

Finally, I am indebted to my wife, Regina, for her constant support and encouragement.

This research was supported in part by the National Sciences and Engineering Research Council of Canada under grant A7634

CONTENTS

	Page
ABSTRACT	v
Chapter 1 - INTRODUCTION	1
Chapter 2 - IMAGE SEGMENTATION TECHNIQUES	4
2.1 Boundary Detection	4
2.2 Characteristic Feature Thresholding	6
Chapter 3 - 3D OBJECT REPRESENTATION MODELS AND SHADING TECHNIQUES	9
3.1 The Cuberille Model	9
3.2 The Triangular Tile Model	9
3.3 Other Models	10
3.4 Techniques Of Displaying 3D Objects on 2D Planes	12
3.4.1 Distance Shading	15
3.4.2 Constant Shading	15
3.4.3 Phong Shading	16
3.4.4 Image-based Contextual Shading	17
3.4.5 Normal-based Contextual Shading	19
3.4.6 Gradient Shading	21
Chapter 4 - NEW TECHNIQUES IN GRADIENT SHADING	22
4.1 The Image-space Gradient Shading Method	25
4.1 The Voxel-based Shading Method	28
4.3 The Normal-averaging Shading Method	39
4.4 Qualitative Analysis of Different Gradient Shading Methods	44
4.5 Experimental Results	51
Chapter 5 - CONCLUSIONS	59
REFERENCES	61

List of Tables

	Page
Table 3.1 The weight W_0 used in the image-based contextual shading method.	18
Table 4.1 A 33 x 33 intensity lookup table for a 16 x 16 x 16 3D image. The light source vector is $L = (3, 2, -1)$. (Each entry is magnified by a factor of 100).	32
Table 4.2 A 33 x 33 intensity lookup table for a 16 x 16 x 16 3D image. (Each entry is magnified by a factor of 100).	36
Table 4.3 A distance matrix of the digital sphere: $x^2+y^2+z^2=16^2$. (-1 indicates invalid distance values since no object is found).	47
Table 4.4a Dot products of $n_T \cdot n_{adj}$. Bold values are values under 96 which implies n_{adj} is farther away from n_T . (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).	48
Table 4.4b Dot products of $n_T \cdot n_{diag}$. Bold values are values under 96 which implies n_{diag} is farther away from n_T . (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).	49
Table 4.4c Dot products of $n_T \cdot n_{avg}$. Bold values are values under 96 which implies n_{avg} is farther away from n_T . (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).	50

List of Figures

	Page
Figure 2.1 Histogram thresholding.	7
Figure 3.1 Diffuse reflection.	13
Figure 3.2a Local coordinate system.	20
Figure 3.2b Object normal, ω , affected by adjacent faces.	20
Figure 3.3 Determination of vector μ . There are 5 possible directions of μ (making angles $\pm 45^\circ$, $\pm \arctan(1/2)^\circ$, and 0° with the positive u -axis (Fig 3.2a)). These angles are determined by the orientation of the adjacent faces.	20
Figure 4.1 Distances of a cross-section of an object from a light source with parallel rays. (The light source here is assumed to be located on the XY-plane).	23
Figure 4.2 Discontinuities in distance due to hidden surfaces.	26
Figure 4.3 A shading intensity table indexed by Δz_x and Δz_y .	30
Figure 4.4a The xz-plane	33
Figure 4.4b The yz-plane	33
Figure 4.5 Dynamic array data structure for intensity look-up table.	35
Figure 4.6a A 3 x 3 portion of the distance matrix. Shaded parts are the centre and its adjacent neighbors.	41
Figure 4.6b A 3 x 3 portion of the distance matrix. Shaded parts are the centre and its diagonal neighbors.	41
Figure 4.7a A 3 x 3 portion of the distance matrix and the corresponding object surface.	41
Figure 4.7b The normal based on adjacent neighbors.	41
Figure 4.7c The normal based on diagonal neighbors.	41
Figure 4.7c The averaged normal.	41
Figure 4.8 (a-f) 3D display of a digital sphere.	54
Figure 4.9 (a-f) 3D display of a liver (coronal view).	55
Figure 4.10 (a-f) 3D display of a liver (coronal view tilted 30° upwards in the vertical plane).	56
Figure 4.11 (a-f) 3D display of a liver (coronal view rotated 15° from patient's right to left).	57
Figure 4.12 (a-f) 3D display of four exposed view of a liver (with cuts at 20, 25, 30, and 35 voxels respectively from the first image plane).	58

Chapter 1 - INTRODUCTION

The availability of medical imaging modalities such as computed tomography scanners, nuclear medicine systems, ultra-sound scanners, digital subtraction angiography systems and the magnetic resonance imaging systems, enable clinicians to view the internal structure of human organs without performing surgery. Such systems produce three-dimensional (3D) information in the form of a series of parallel two-dimensional (2D) images, each of which displays a cross-section of the internal structure. Inspecting the 2D images one after the other does not give clinicians a full understanding of the shape of the objects they wish to view. With the aid of computer imaging devices, 3D images of objects can be produced. The display of 3D data usually involves the following three steps:

- (1) image segmentation,
- (2) merging of 2D data into a 3D representation,
- (3) display of the 3D object .

Image segmentation can be regarded as a preprocessing step being applied to the data. It involves grouping parts of a generalized image into units that are homogeneous with respect to one or more characteristics. Very often the object of interest is buried inside the objects of non-interest. For instance, a physician interested in a bone study would like the soft tissue data removed from the CT (Computed Tomography) images. One aspect of image segmentation frequently used is a global thresholding technique. The first chapter of this thesis briefly mentions some segmentation techniques based on thresholding and boundary detection.

Once the images are segmented, the information can be used to produce a data structure (according to some specified data model) so that the 3D

objects can be represented. There are two main approaches for 3D reconstruction of objects from 2D slices: the *volume-oriented* approach and the *surface-oriented* approach. In the volume-oriented approach the 2D slices are combined to form a 3D volume. The smallest element in the volume is a *voxel* whose size in two dimensions is equal to the size of a CT pixel. The object is then depicted by the *opaque* voxels in the volume. In the surface-oriented approach the boundaries of the objects are connected together into a mosaic of polygonal elements, usually triangles, which represent the surfaces. In the second chapter of this thesis approaches towards both techniques as well as other techniques are discussed.

Having the 3D representation of the object defined, one can then visualize the object by projecting the 3D objects onto a 2D screen. Among these projections are wire-framing, particle-framing and shading. Most of the display techniques are based on shading techniques developed in computer graphics; the brightness (shading intensity) of a point on the surface of an object depends upon three factors: (1) the intensity of the ambient light, (2) the amount of diffuse reflection and (3) the amount of specular reflection (usually ignored in medical imaging). In Chapter 2, several display techniques based on the aforementioned data models are presented.

The emphasis of this thesis is on the display of 3D data using shading techniques. The main contributions of the thesis are: (1) The introduction of an efficient shading technique (voxel-based gradient shading) and (2) the introduction of a high quality shading technique (normal-averaging gradient shading). This thesis is divided into two main parts. Part 1 (Chapters 2 and 3) gives a brief discussion on image segmentation, 3D object representation techniques and previous research effort on 3D shading techniques. Part 2 (Chapter 4) presents three gradient shading methods; one recently developed

by the author, one proposed by Gordon and Reynolds [18] and a new shading technique (which is based on the framework proposed by Gordon and Reynolds). This new technique produces the best quality images among the three methods mentioned. Sections 4.4 and 4.5 provide a qualitative analysis and a verification of the results obtained.

Chapter 2 - IMAGE SEGMENTATION TECHNIQUES

The first step involved in the display of 3D data is image segmentation. Image segmentation is the division of an image into different regions, each having certain properties. In the past decade, many image segmentation techniques have been proposed [12]. These segmentation techniques, according to Fu and Mui [12], can be divided into three categories: (1) boundary detection, (2) characteristic feature thresholding and (3) region extraction. Since most biological and medical applications of image segmentation concentrate in the first two categories, only the first two categories will be considered here.

2.1 Boundary Detection

Boundary (edge) detection is a segmentation technique based on the detection of discontinuities. A boundary is found when there is a more or less abrupt change in gray level. Experiments with the human visual system show that boundaries in images are extremely important; often an object can be recognized from only a rough outline. In recent years, many boundary detection techniques have been proposed, some examples are presented in [1, 3, 9, 12, 26, 29, 34, 39], many of them are 2D edge detection techniques and some of which can be adapted to the 3D environment. The techniques in their original 2D form will be described. Since edge detection is an area which has attracted tremendous attention in the past, it is impossible to cover all the previous research efforts in this section. The following will present some fundamentals and also mention some of the recent research effort.

Early work on edge detection primarily used first- and second-derivative operators for performing the two-dimensional differentiation on

images. A change of intensity in the image gives rise to a peak in the first derivative of intensity change, or a zero crossing in the second derivative. In the case of a first derivative operator, the gradient function is often used,

$$\nabla f(x,y) = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]. \quad (2.1)$$

The digital approximation to the gradient is,

$$\nabla_x f(x,y) = f(x+1,y) - f(x,y), \quad (2.2a)$$

$$\nabla_y f(x,y) = f(x,y+1) - f(x,y). \quad (2.2b)$$

An edge point then will be a point where the value

$$|\nabla f(x,y)| = \max(|\nabla_x f(x,y)|, |\nabla_y f(x,y)|). \quad (2.1)$$

is greater than a certain threshold value. Examples of the first derivative edge detector are the Roberts, Prewitt and Sobel operators. In the case of a second derivative operator, the Laplacian function is often used. In digital form, it is given by

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y-1) + f(x,y+1) - 4f(x,y). \quad (2.3)$$

which is the digital convolution of f with

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The digital Laplacian, since it is a second-derivative operator, has zero response to linear ramps, but it responds to the "shoulders" at the top and bottom of a ramp, where there is a change in the rate of change of gray level. Rosenfeld [36] gives a more detailed analysis of the above 1st and 2nd derivative operators. Second derivative operator responds more vigorously to high frequency signal than first derivative operator. Thus in a noisy image (characterized by high frequency signals), a second derivative operator will enhance noise more than a first derivative operator. Therefore, the first

derivative operator is, in general, a better operator than a second derivative operator in edge detection. Since either the 1st and 2nd derivative operators will enhance edges as well as noise, a smoothing operation is usually employed before edge detection is done. Recent research effort in edge detection usually employ a median filter (or a variant of it [30]), and Gaussian smoothing function [3, 26] before applying the derivative operators.

Edge (boundary) detection is mostly used in 2D image processing rather than 3D applications due mainly to their complexity (as compared to simpler techniques such as thresholding) and their computational inefficiency, although some effort has occurred in the past to apply boundary detection in 3D CT data [29]. The following section briefly describes a global thresholding technique which is often used in 3D image segmentation.

2.2 Characteristic Feature Thresholding

Thresholding is a technique commonly used in image segmentation especially in 3D medical data, due mainly to its simplicity and its ability to single out features easily with a high degree of homogeneity. In its most general form, thresholding can be described mathematically as:

$$s(x, y, z) = k \quad \text{if} \quad T_{k-1} \leq f(x, y, z) < T_k \quad (2.4)$$

$$k = 1, 2, 3 \dots m.$$

where

(x, y, z) are the x , y , and z coordinates of a voxel,

$s(x, y, z)$ is the segmented function of (x, y, z) ,

$f(x, y, z)$ is the characteristics feature (e.g. gray-level function of (x, y, z) ,

$T_0 \dots T_m$ are threshold values (T_0 and T_m are the minimum and maximum value respectively),

m is the number of distinct labels (divisions) assigned to the segmented images.

A thresholding operator T can be viewed as a testing function of the form

$$T(x, y, z, P(x, y, z), f(x, y, z))$$

where

$P(x, y, z)$ denotes some local property of the point (x, y, z) (e.g. the average gray level over some neighborhood).

Thresholding can be divided into three types depending on the functional dependencies of the threshold operator T :

- (1) Global threshold: T depends only on $f(x, y, z)$,
- (2) Local threshold: T depends on both $f(x, y, z)$ and $P(x, y, z)$
- (3) Dynamic threshold: T depends on all parameters stated.

Since image segmentation is not the main interest of this thesis, only the more commonly used global thresholding techniques will be discussed. Two papers Weszka [46] and Sahoo et. al. [37] summarize much of the past research effort in thresholding.

The best way to pick a global threshold T is to search the histogram of the image. A gray-level histogram of an image is a function that gives the frequency of occurrence of each gray level in the image. When the gray levels are quantized from 0 to n , the value of the histogram at a particular gray level g , denoted by $h(g)$, is the number or fraction of voxels with that gray level. If the histogram is bimodal, the threshold will be the minimum separating the two peaks as shown in Fig. 2.1.

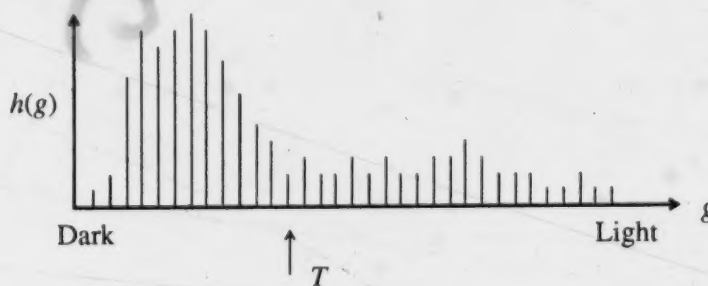


Fig. 2.1 Histogram thresholding.

The single-threshold method is useful in simple situations, but problems arise when the image background has a varying gray level or when gray level regions vary smoothly by more than the threshold. Chow and Kaneko [6] improved the thresholding technique by employing a spatially varying threshold. The technique divides the image into rectangular subimages and computes a threshold for each subimage. A subimage may not have a threshold when its histogram is not bimodal. In such cases thresholds will be calculated by interpolating values from neighboring subimages that are bimodal. The whole image is then segmented by using separate thresholds for each subimage. Recent research efforts in 3D imaging [15, 16, 21, 27] mostly employ global thresholding techniques for image segmentation due mainly to its simplicity and its ability to easily separate homogeneous features such as soft tissues from bone and vice versa. The 3D images shown in Figs 9-12 of Chapter 4 are segmented using the global thresholding technique described above. First, an intensity histogram is generated for the entire 3D object and a threshold value is roughly obtained from the histogram. Then, a manual adjustment is applied to obtain a threshold value which gives the best segmentation on average out of the 64 slides of 2D images.

Chapter 3 - 3D OBJECT REPRESENTATION MODELS AND SHADING TECHNIQUES

In order that detected objects can be visualized, methods have to be developed to represent the extracted surface such that they can be displayed on a graphics device. A number of surface representation models have been proposed. The following discusses several 3D surface representation methods. This chapter also describes the shading technique based on the *cuberille* representation model.

3.1 The Cuberille Model

A model proposed by Herman and Liu [24] represents 3D surfaces by means of a *cuberille*. A *cuberille* is a partitioning of space into equal cubes by three orthogonal sets of equally spaced parallel planes. The volume elements (*voxels*) become components of a *cuberille* S and all surface elements are faces of S . Herman and Liu [24] proposed a constant shading method that can be applied to the *cuberille* model. This method will be briefly described later in this chapter. In addition, a number of display techniques, based on the *cuberille* representation, have been proposed by Chen and Herman et al. recently [4]. Such techniques will also be discussed later in this chapter.

3.2 The Triangular Tile Model

Another approach to represent 3D surface is by means of polygonal patches. A variety of polygons can represent surface of the reconstructed structure; the most popular and simplest of these is the triangle. A number of approaches have been proposed. The following describes some of them.

One approach, proposed by Keppel [28] is a triangular tile technique for reconstructing a surface between contours (parallel planes of edges) by searching for an optimal tile arrangement with a graph theoretic approach. The algorithm first decomposes the contours into convex and concave subsets, then arranges the triangular tiles by maximizing the volume enclosed by convex surfaces. The volumes of the tetrahedrons corresponding to each possible triangular face are associated with a directed graph; the arrangement of the triangles can be found by finding the maximum cost path of the graph.

Another approach, proposed by Fuchs et al. [14], employs a fast algorithm for finding the path that corresponds to a tile arrangement by finding the minimum cost path of the graph. Their algorithm approximates the surface by minimizing the surface area of the triangles.

A third approach, proposed by Christiansen and Sederberg [8], suggests an algorithm that produces the triangular tiles by mapping the adjacent planes of contours onto unit squares. Then the triangle arrangement was selected on the basis of the shortest diagonal, with the requirement that the first nodes of each band must approximate one another. This method is easily adapted to approximate surface that does not form simple closed edges in cross-sections.

When 3D surfaces are represented by triangular tiles (or in general, polygonal tiles), common display techniques such as Gouraud shading or Phong shading are used for display.

3.3 Other Models

A different approach, adopted by Wu, Abel and Greenberg [48], gives rise to a technique which reconstructs surface with a spline approximation. In

this method, B-splines represent the sectional contours and cardinal splines interpolate between the contour planes.

Yet another approach that should be mentioned is *Octree-Encoding* [17, 32]. It is a geometric modeling technique in which 3D objects can be represented to any specified resolution (up to the resolution of the original image) in a hierarchical 8-ary tree structure or *octree*. Each node in the tree corresponds to a region of the whole object and has one or more values which define the region. If the value of the node completely describes the region, it is a terminal node or a leaf. If not, an ambiguity exists and the node points to the eight children that represent the eight subregions or octants of the parent node. To display the object, a perspective transformation and a shaded graphic technique is used.

The simplest form, perhaps, to represent a 3D object is by means of the original object space -- a volume. Each element of the volume is a *voxel*. The object is then depicted by all *non-transparent* voxels (i.e. voxels with non-zero intensity values). This representation is called the *voxel model*. The voxel model and the cuberille model mentioned above essentially represents the same thing -- a cubic volume in 3D space. They differ mainly in the application of shading techniques in either model. In the voxel model, shading is determined by the location of the centre of the voxel and neighboring voxels. In the cuberille model, however, shading is determined by the orientation of the faces (surface elements of the set S as described in [24]) and faces on neighboring cuberilles. The following section gives a general discussion on the formation of shading intensity and shading techniques applied on the cuberille model.

3.4 TECHNIQUES OF DISPLAYING 3D OBJECTS ON 2D PLANES

Displaying 3D objects on a 2D plane requires depth-cues to create the illusion of the third dimension. Wire-framed objects usually use hidden surface removal or motion parallax to create a feeling of 3D. Other methods use shading to simulate both the object surface characteristics and the position and orientation of the surfaces with respect to the light sources. In general terms, the shading value or the intensity of a point on a surface depends on three factors: (1) the amount of ambient light, (2) the amount of diffuse reflection and (3) the amount of specular reflection. Thus the equation for the shading intensity of a point on the surface can be written as:

$$I = I_a + I_d + I_s \quad (3.1)$$

where

I_a = the amount of ambient light,

I_d = the amount of diffuse reflection, and

I_s = the amount of specular reflection.

In medical applications, the term I_s is not often used due mainly to the fact that it is computationally expensive and that I_d gives sufficiently good quality images. Thus in a medical application

$$I = I_a + I_d \quad (3.2)$$

The amount of diffuse reflection depends on the cosine of the angle between the incident ray and the surface normal and the distance from the point light source to that particular point on the surface, see Fig 3.1.

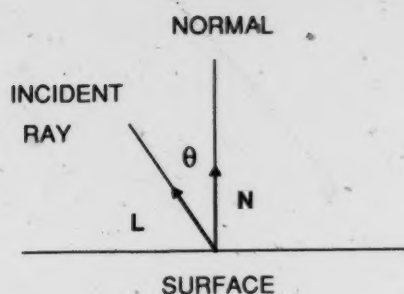


Fig 3.1 Diffuse reflection.

Thus

$$I_d = f(\cos \theta, d) \quad (3.3a)$$

where

θ = angle between the incident light ray and the surface normal,

d = distance from the light source to the surface.

The value of $\cos \theta$ can be calculated by obtaining the dot (scalar) product of the unit normal vector and the unit vector of a line parallel to the light ray. Therefore,

$$\cos \theta = \mathbf{n} \cdot \mathbf{l}, \text{ and} \quad (3.3b)$$

$$I_d = f(\mathbf{n}, \mathbf{l}, d) \quad (3.3c)$$

where

$\mathbf{n}$ = unit vector normal of the surface,

$\mathbf{l}$ = unit vector along the line from the surface to the light source.

In theory, illumination is inversely proportional to the square of the distance from a point light source to the surface. In practice, the factor $1/d^2$ does not produce natural looking images, so $1/d$ is usually used. When all the above-mentioned factors are taken into account, Gordon and Reynolds [18] state that

the following expression yields good results:

$$I = \frac{(I_{max} - I_a)(D - d)(\cos \theta)^p}{D} + I_a \quad (3.4)$$

where

I_{max} = the maximum intensity,

I_a = the ambient light,

D = the distance at which the illumination falls to zero (in practice, D may be considered to be the diameter of the smallest sphere which encloses the object),

d = the distance from the light source to the surface,

θ = angle between the surface normal and the incident ray,

p = empirically determined parameter which adjust the angular dependence on the $\cos \theta$ term.

The value p varies between 0 and 1 and, according to [18] the value $p=0.2$ produces good images. Chen and Herman et al. [4] suggest a more general form which can be used to describe many shading methods (some of which will be discussed below). The formula is as follows:

$$S(P) = \left[\frac{M - L}{2R} (2R - d) N(\theta) + L \right]_L^M \quad (3.5a)$$

where

$$\left[V \right]_L^M = \begin{cases} L, & \text{if } V < L, \\ V, & \text{if } L \leq V \leq M, \\ M, & \text{if } M < V. \end{cases} \quad (3.5b)$$

In this formula, P is a point on the surface of an object and $S(P)$ is the shading value of point P . M is the maximum intensity that can be produced by the

display hardware and L is the amount of ambient light. d is the distance of P from the source of light and θ is the angle between the direction of light and the estimated surface normal at P . R is the radius of the smallest sphere that encloses the object. $N(\theta)$ is a function of θ which simulates the properties of a diffuse reflecting surface. In many of the following shading formulations, let $N(\theta)$ be a function of $\mathbf{n}$ and $\mathbf{l}$, i.e.

$$N(\theta) = \cos \theta = f(\mathbf{n}, \mathbf{l}). \quad (3.6)$$

3.4.1 Distance Shading

The simplest shading method, perhaps, is distance shading. This form of shading does not require the complex process of estimating the object normals of surfaces therefore it is very computationally efficient. The formula for distance only shading can be obtained by setting $N(\theta)$ in Equation 3.5 to 1. In the voxel-based environment, d is the distance from the centre of the voxel to the light source. Distance shading produces smooth looking images due to the fact that the distance d for neighboring faces does not differ too much. In a practical sense, however, this is a handicap, because minute textural information will be lost. Another difficulty is that distance shading does not take into account surface orientation; faces on either side of an edge have similar distance from the light source. This is an important drawback because real-edge information cannot be detected.

3.4.2 Constant Shading

A simple shading method frequently used during the emerging stage of computer graphics is constant shading. This form of shading can be applied to objects with surfaces represented by a polygonal mesh. The shading value on each polygon is constant and is estimated in terms of the surface normal of the

polygon and the direction of the light source. A not-so-recent implementation of constant shading in the cuberille environment has been done in [24] by Herman and Lui. In such an implementation, the distance shading term is not used, i.e. the $(2R - d) / 2R$ term in Equation 3.5 is not used. This eliminates the smoothing effect of the distance shading. In [24] Herman and Liu implemented a low-pass filter to smooth out the "squareness" produced by this method. In [4] Chen and Herman re-worked this technique by introducing the distance shading term as in Equation 3.5. If Equation 3.5 is used, $N(\theta)$ will be in the form

$$N(\theta) = \cos \theta \quad (3.7)$$

where θ is the angle between the surface normal of the polygon and the direction of the light source. Chen and Herman [4] suggest a more general form

$$N(\theta) = \left[\cos \left(\frac{\theta}{n} \right) \right]^p \quad (3.8)$$

and recommend, after extensive experiments, that $p=0.6$ and $n=2$. A major problem with constant shading is the strong appearance of artificial edges at certain rotation angles. When compared with distance shading, however, this method is still preferable since more detail of the objects can be visualized.

3.4.3 Phong Shading

In the 1970's two smooth shading methods revolutionized the world of computer graphics: intensity interpolation, or Gouraud shading and normal vector interpolation, or Phong shading. Phong shading is computationally more expensive than Gouraud shading, but is visually preferable since it is less susceptible to the Mach-band effect. In the world of medical imagery, Phong shading is regarded as the *de facto* standard for image quality. Since

the principles of Gouraud and Phong shadings are well known, they will not be discussed in detail, only Phong shading will be discussed briefly in the following.

In the normal vector interpolation technique proposed by Bui-Tuong Phong [35], obtaining the shading value requires calculating the resultant normal vector at each vertex where polygons meet. This vertex normal is then interpolated linearly across the polygon surface. The interpolation is carried out in two stages. First, the normal vector components are interpolated along the polygon edges joining the vertices to find the values at the beginning and end of each scanline; then the components are interpolated along the scanline to estimate the object normal for each point of the image.

3.4.4 Image-based contextual shading

In an attempt to overcome the problems caused by artificial edges in the cuberille environment [24], Chen and Herman [4] proposed the image-based contextual shading method. This method tries to eliminate artificial edges caused by constant shading. The idea is to assign a shading value to points on a face that depends not only on the orientation of the face itself, but also on the orientation of adjacent faces (the four faces which share an edge with the original face). Denote the angle between the normal to a face f and the direction of light as θ_0 and let θ_1 , θ_2 , θ_3 and θ_4 be similarly defined angles for the four adjacent faces. Then the shading value of all points on the face f is determined by Equation 3.5 evaluated at the centre point of f with

$$N(\theta) = \sum_{i=0}^4 w_i \left[\cos\left(\frac{\theta_i}{n}\right) \right]^p, \quad \sum_{i=0}^4 w_i = 1. \quad (3.9)$$

	Number of visible critical faces in the neighborhood				
	0	1	2	3	4
Critical	1/16	1/8	1/4	1/2	1
Noncritical	1	15/16	7/8	3/4	1/2

Table 3.1 The weight W_0 used in the image-based contextual shading method.

The weights W_i depend on whether the face f is *critical* or *non-critical*, and on the number of visible adjacent faces which are critical. A critical face is a face whose normal is less than 45° from the light source vector, and the rest are called non-critical faces. Herman and Udupa proposed a unique scheme of assigning weights. To describe this, first let n_c be the number of *visible* critical faces in the neighborhood and also define a weighting function for adjacent faces

$$W_i = \begin{cases} (1 - W_0) / n, & 1 \leq i \leq 4, n > 0 \\ 0 & 1 \leq i \leq 4, n = 0 \end{cases} \quad (3.10)$$

where n is the number of *visible* critical faces adjacent to f if f is non-critical; otherwise, n is the number of *visible* non-critical faces adjacent to f .

Now the rules are:

(1) If f is critical then

- (a) assign zero weight to adjacent critical face(s),
- (b) look-up from Table 3.1 the W_0 for f being critical (1st row) and n non-critical faces,
- (c) substitute this W_0 into Equation 3.10 and get W_i for non-critical adjacent faces, and
- (d) finally substitute W_0 and W_i 's and θ_i 's into Equation 3.9 and obtain $N(\theta)$ for face f .

(2) If f is non-critical then

- (a) assign zero weight to adjacent non-critical face(s),
- (b) look-up from Table 3.1 the W_0 for f being non-critical (2nd row) and n critical faces,
- (c) substitute this W_0 into Equation 3.10 and get W_i for critical adjacent faces, and
- (d) finally substitute W_0 and W_i 's and θ_i 's into Equation 3.9 and obtain $N(\theta)$ for face f .

The image-based shading method has been used mainly in the cuberille environment. In such an implementation, a one byte neighbor-code is associated with each face of the cuberille (there are only 81 possible arrangements of the faces (ref. Fig. 3.3), since 9 for each axis, and $9 \times 9 = 81$). Since there are only 81 possible values of $N(\theta)$, these values can be stored in a look-up table. Thus contextual shading is no more expensive than constant shading, and yet produces better images. However, because of the arrangement of critical and non-critical faces, the surface intensity over the whole image varies unnaturally with the direction of light.

3.4.5 Normal-based contextual shading

In order to solve the rotational invariant problem of the image-based shading method, Chen and Herman [4] propose a similar technique called the normal-based contextual method. The implementation of this method uses the same adjacent faces arrangement and the same neighbor-code data structure. However, instead of using weighting factors to calculate $N(\theta)$, object normals are calculated based on the orientation of a face f as well as the orientation of the adjacent faces. The object normal, ω , of a centre point P of a face f is

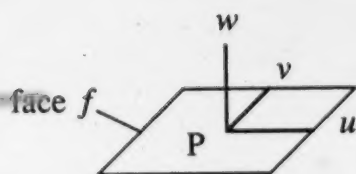


Fig. 3.2a Local coordinate system.

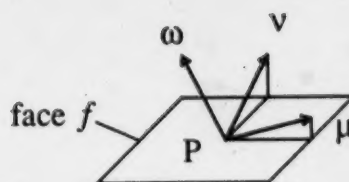


Fig. 3.2b Object normal, ω , affected by adjacent faces.

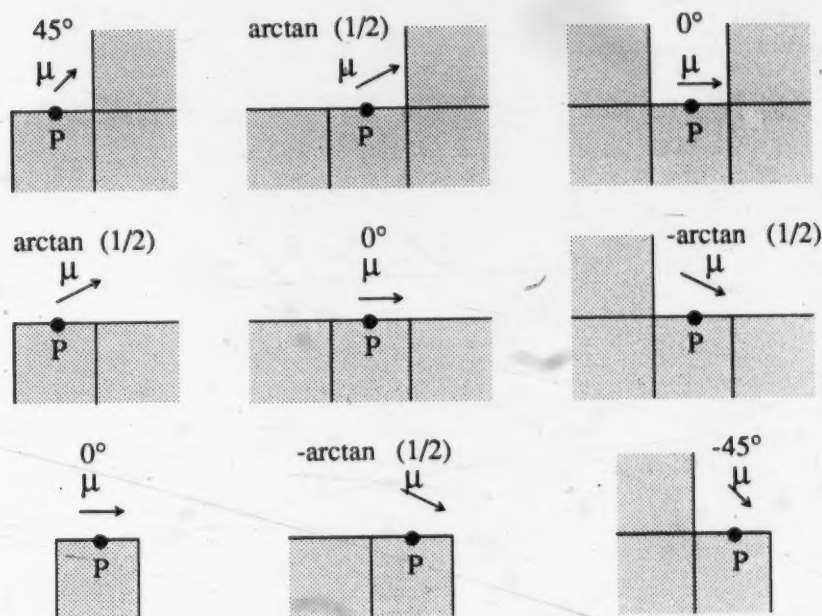


Fig. 3.3 Determination of vector μ . There are 5 possible directions of μ (making angles $\pm 45^\circ$, $\pm \arctan(1/2)$, and 0° with the positive u -axis (Fig 3.2a)). These angles are determined by the orientation of the adjacent faces.

defined as the cross product of vectors μ (in the u - w plane) and v (in the v - w plane), see Figs 3.2a and 3.2b. The vectors μ and v can each have (independently) one of five possible directions, depending on the adjacent faces in the positive and negative u -directions (for μ) and v -directions (for v). The exact method by which μ is determined is illustrated in Fig. 3.3 for each of the nine arrangements of adjacent faces in the positive and negative u -

directions. A similar method can be used to obtain v . With the above algorithmic changes, the normal-based method solved the rotational invariant problem of the image-based shading method; that is, the texture pattern does not change appreciably as the object is rotated. This method of shading does produce good images, and the quality is found to be comparable with the Phong shading method, but with a considerable improvement in computation speed.

3.4.6 Gradient Shading

With the exception of distance shading, the afore-mentioned shading methods may be described as *object-space shading* methods due to the fact that extra data (face-orientation, vertex normals, or a neighbor-code) need to be stored with the object representation. Object-space methods have the disadvantage that the normals must be recalculated if the object is modified. For systems allowing rapid interaction with the object (e.g. in surgical planning), this problem is of great importance. To avoid this problem two methods have been proposed: (1) the voxel-based method [10] recently developed by the author and (2) the image-space shading method proposed by Gordon and Reynolds [18]. Both methods can be termed as a *gradient shading* method since both methods make use of the gradient vectors of the object surfaces to find the shading values. These methods will be discussed in detail in the next chapter and a new display technique is proposed that is based on the image-based gradient shading method. This technique produces images of higher quality than that produced by Gordon and Reynolds. The new method is called the *normal-averaging gradient shading method*.

Chapter 4 - NEW TECHNIQUES IN GRADIENT SHADING

In this chapter three gradient shading methods will be presented. The first one is called the *image-space gradient shading method* proposed by Gordon and Reynolds [18]. The second was developed at an earlier stage of this research and was developed independently of the *image-space gradient shading method* although similar results are obtained. This method is called the *voxel-based method* and an abstract of this method has been reported in [10]. The voxel-based method is aimed at producing images by trading image quality in favour of speed. However when compared with the image-space gradient shading method, it produces images of comparable quality. The third method is called the *normal-averaging method*. This is a newly developed technique and is found to produce the best quality images among the three methods at the expense of a slight increase in execution time.

Unlike the object-space shading methods which require extra data to be stored with the object representation, the gradient shading method uses no more data than a complete pre-image obtained by coordinate transformation, hidden-surface removal and scan-conversion. One suitable pre-image is the *Z-buffer* used for hidden-surface removal. The Z-buffer contains, for each pixel, the distance from the light source to the closest point on the object which projects onto that pixel. The Z-buffer, or in other words, the *distance matrix*, then contains a complete representation of the visible surface of the object of the form $z = z(x,y)$, see Fig. 4.1. The object normal $\mathbf{N}$ is then computed for every point in terms of the gradient vector ∇z at that point. From the normal $\mathbf{N}$, the light direction $\mathbf{L}$, the distance, and the radius of the minimal spanning sphere, the shading intensity of a point on the object can be easily obtained using Equation 3.5. Since the distance information is the only data needed, any other technique besides the Z-buffer algorithm, can be used to obtain a

distance matrix.

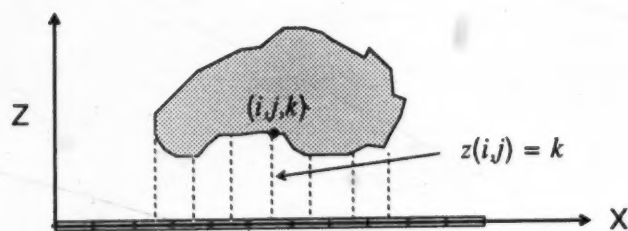


Fig. 4.1 Distances of a cross-section of an object from a light source with parallel rays. (The light source here is assumed to be located on the XY-plane).

All of the three methods: the image-space gradient method proposed by Gordon and Reynolds [18], the earlier proposed voxel-based method, and the newly proposed normal-averaging method require the distance matrix as the pre-image. Given an object volume of $n \times n \times n$ voxels, an $M \times M$ distance matrix can be generated. Each entry of the matrix can be represented by

$$z_{ij} = z(x=i, y=j), \quad (1 \leq i \leq M, 1 \leq j \leq M).$$

Here the value of M depends on the rotation of the $n \times n \times n$ object volume in 3D space. Since the maximum projected dimension (isometric projection only) of the object is $\sqrt{3}n$, therefore the maximum value of M would be the integer value of $\sqrt{3}n$ plus 1. Thus $n \leq M \leq \text{int}(\sqrt{3}n) + 1$. From the distance matrix the normal at any point of the object surface $z = z(x, y)$ can be obtained from the gradient vector

$$\nabla z = \mathbf{N} = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1 \right).$$

The image-space gradient shading method and the voxel-based method differ in the way $\partial z / \partial x$ and $\partial z / \partial y$ are obtained and the way the shading intensities are estimated. The normal-averaging method obtains the shading intensities by means of ∇z and $\nabla z'$ (where $\nabla z'$ is the gradient vector obtained from the

diagonal neighbours in the distance matrix). The following sections describe the image-space gradient shading method and explain the two proposed methods in detail.

4.1 The Image-Space Gradient Shading Method

In early 1985 Gordon and Reynolds produced a paper [18] on a 3D shading technique called the *image-space gradient shading method*. This method suggests estimating the surface normal by the gradient vector ∇z by

$$\nabla z = \mathbf{N} = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1 \right).$$

Where $\partial z/\partial x$ is the surface gradient with respect to the x-axis and $\partial z/\partial y$ is the surface gradient with respect to the y-axis. $\partial z/\partial x$ and $\partial z/\partial y$ are estimated from values of the distance matrix in terms of the backward, forward or the central difference. To obtain $\partial z/\partial x$ (a similar way can be used to obtain $\partial z/\partial y$), Gordon and Reynolds [18] proposed the following schema for the backward, forward and central differences:

backward difference

$$\delta_b = z_{ij} - z_{i-1j}, \quad (2 \leq i \leq M, 1 \leq j \leq M), \quad (4.1a)$$

forward difference

$$\delta_f = z_{i+1j} - z_{ij}, \quad (1 \leq i \leq M-1, 1 \leq j \leq M), \quad (4.1b)$$

central difference

$$\delta_c = \frac{1}{2} (z_{i+1j} - z_{i-1j}), \quad (2 \leq i \leq M-1, 1 \leq j \leq M). \quad (4.1c)$$

The distance matrix does allow rapid calculation of the surface normal and hence the intensity values. However, it does not give information about surface continuities. In cases where there are discontinuities due to hidden surfaces (see Fig. 4.2), the quality of the image degraded due to the fact that the object normal is incorrect at discontinuities. To solve the problem of discontinuities between objects, Gordon and Reynolds [18] proposed a

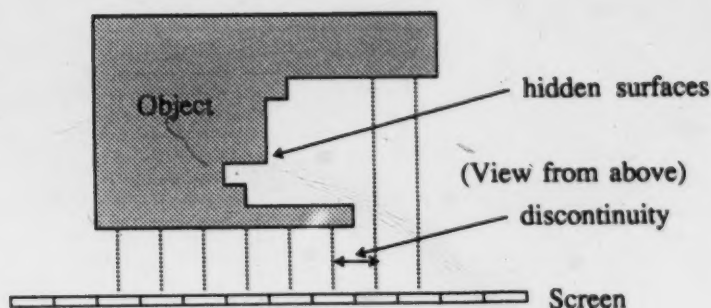


Fig. 4.2 Discontinuities in distance due to hidden surfaces.
(Figure extracted from p. 365 of [18]).

weighted average function as follows,

$$\frac{\partial z}{\partial x} = \frac{W(|\delta_b|) \cdot \delta_b + W(|\delta_f|) \cdot \delta_f}{W(|\delta_b|) + W(|\delta_f|)}, \quad (4.2)$$

where, for $\delta \geq 0$, $b > a$

$$W(\delta) = \begin{cases} 1, & \text{if } \delta \leq a, \\ \epsilon, & \text{if } \delta \geq b, \\ \frac{1+\epsilon}{2} + \frac{1-\epsilon}{2} \cos\left(\frac{\delta-a}{b-a} \pi\right), & \text{otherwise.} \end{cases} \quad (4.3)$$

A similar expression is used for $\partial z / \partial y$. ϵ , a and b are empirical constants: ϵ is a small number large enough so that division by 2ϵ maintains sufficient precision. In the implementation described in [18], $\epsilon=10^{-5}$ is used. a and b are parameters which control the points at which the δ 's change their weights: a is the maximal δ at which the large weight ($W=1$) is given, and b is the minimal δ at which the small weight ($W=\epsilon$) is given. For δ 's between a and b the weight varies continuously with the cosine function. This weight function minimizes discontinuities in the image when the orientation of the object is changed. This formula ensures that small δ 's are given large weights and large δ 's are given small weights. If $W(|\delta_b|)$ equals $W(|\delta_f|)$ then the central difference is obtained;

this will be the case independent of the values of $W(|\delta_b|)$ (or $W(|\delta_d|)$). The parameters a and b ensure that the appropriate normal is obtained for rapidly changing slopes. The shading intensity is obtained by substituting

$$N(\theta) = \frac{\mathbf{N}}{|\mathbf{N}|} \cdot \frac{\mathbf{L}}{|\mathbf{L}|}$$

in Equation 3.5. Gordon and Reynolds [18] found that $a=2$, $b=5$ produces good images.

4.2 The Voxel-Based Shading Method

The voxel-based method is designed for efficient display of 3D objects without sacrificing image quality. Like the image-space gradient shading method, it makes use of the distance matrix to estimate surface normals. To obtain the shading value using Equation 3.5 as described in Chapter 3, the voxel-based method assumes $N(\theta) = \cos \theta = \mathbf{n} \cdot \mathbf{l}$ (where $\mathbf{n}$ and $\mathbf{l}$ are unit vectors of the surface normal, $\mathbf{N}$, and the light source vector, $\mathbf{L}$). The efficiency comes from the fact that the values of $N(\theta)$ are pre-generated, and that during the intensity calculations, the $N(\theta)$ can be obtained by a simple table look-up. The following describes how the look-up table is derived. The object normal of a point on the surface of the object can be represented by

$$\begin{aligned} \mathbf{N} &= \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1 \right) \\ &\approx \left(\frac{\Delta z_x}{\Delta x}, \frac{\Delta z_y}{\Delta y}, -1 \right). \end{aligned}$$

As mentioned earlier, the voxel-based method is developed independently of the image-space gradient shading method although similar results are obtained. The voxel-based method differs from the image-space gradient shading method by introducing the use of intensity LUT, excluding equation 4.2 and 4.3, and obtaining the backward, forward and central differences in a different manner. The image-space gradient shading method obtains differences from adjacent entries in the distance matrix, i.e. assume $\Delta x = \Delta y = 1$. Here, the voxel-based method uses $\Delta x = \Delta y = 2$. Thus

$$\mathbf{N} = \left(\frac{\partial z}{2}, \frac{\partial z}{2}, -1 \right)$$

To get Δz_x (the difference in depth in the x -direction), the following schema is proposed, for an $M \times M$ distance matrix:

backward difference

$$\Delta z_{xb} = \begin{cases} \min [2(z_{ij} - z_{i-1j}), n] & \text{if } (z_{ij} - z_{i-1j}) \geq 0 \\ \max [2(z_{ij} - z_{i-1j}), -n] & \text{if } (z_{ij} - z_{i-1j}) < 0 \end{cases} \quad (4.4a)$$

and $(2 \leq i \leq M, 1 \leq j \leq M)$,

forward difference

$$\Delta z_{xf} = \begin{cases} \min [2(z_{i+1j} - z_{ij}), n] & \text{if } (z_{i+1j} - z_{ij}) \geq 0 \\ \max [2(z_{i+1j} - z_{ij}), -n] & \text{if } (z_{i+1j} - z_{ij}) < 0 \end{cases} \quad (4.4b)$$

and $(1 \leq i \leq M-1, 1 \leq j \leq M)$,

central difference

$$\Delta z_{xc} = \begin{cases} \min [z_{i+1j} - z_{i-1j}, n] & \text{if } (z_{i+1j} - z_{i-1j}) \geq 0 \\ \max [z_{i+1j} - z_{i-1j}, -n] & \text{if } (z_{i+1j} - z_{i-1j}) < 0 \end{cases} \quad (4.4c)$$

and $(2 \leq i \leq M-1, 1 \leq j \leq M)$,

Here again n is the length (in voxels) of a side of an object volume. The reason for the *min* and *max* function in Δz_{xb} , Δz_{xf} and Δz_{xc} is that their values may get smaller than $-n$ or larger than n . In all cases, the *min* and *max* function will limit the values to be $-n \leq \Delta z_{xb}, \Delta z_{xc}, \Delta z_{xf} \leq n$. The limiting functions are bound to introduce errors in the estimation of surface normal, however it is extremely rare that an object has such a steep surface gradient within a distance of approximately two voxels. Let the light source vector be $\mathbf{L} = (l_x, l_y, l_z)$, it follows that

$$\begin{aligned} N(\theta) &= \frac{\mathbf{N}}{|\mathbf{N}|} \cdot \frac{\mathbf{L}}{|\mathbf{L}|} \\ &= \frac{\left(\frac{\Delta z_x}{2}, \frac{\Delta z_y}{2}, -1 \right) \cdot (l_x, l_y, l_z)}{\left| \left(\frac{\Delta z_x}{2}, \frac{\Delta z_y}{2}, -1 \right) \right| \cdot \left| (l_x, l_y, l_z) \right|} \end{aligned}$$

$$= \frac{\frac{\Delta z_x}{2} l_x + \frac{\Delta z_y}{2} l_y - l_z}{\left\{ \left[\left(\frac{\Delta z_x}{2} \right)^2 + \left(\frac{\Delta z_y}{2} \right)^2 + 1 \right] \cdot \left[l_x^2 + l_y^2 + l_z^2 \right] \right\}^{1/2}}$$

$$N(\theta) = \frac{\Delta z_x l_x + \Delta z_y l_y - 2 l_z}{\left\{ \left[(\Delta z_x)^2 + (\Delta z_y)^2 + 4 \right] \cdot \left[l_x^2 + l_y^2 + l_z^2 \right] \right\}^{1/2}} \quad (4.5)$$

Given that the direction of the light source is constant for a given view, then

$$N(\theta) = f(\Delta z_x, \Delta z_y).$$

Since the value of $N(\theta)$ depends only on two parameters which are finite and within a small bound, a look-up table can be defined (for values of $N(\theta)$) with rows indexed by Δz_x and columns indexed by Δz_y as shown in Fig. 4.3.

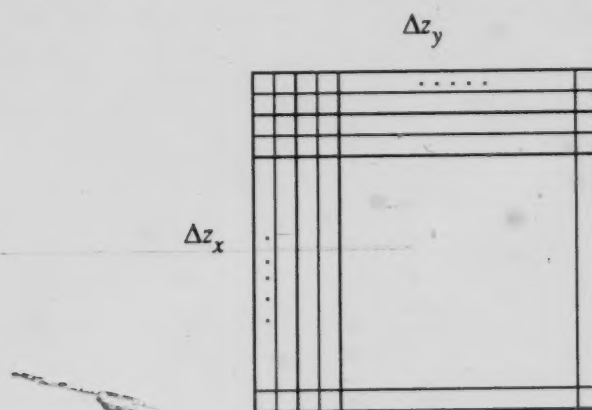


Fig. 4.3 A shading intensity table indexed by Δz_x and Δz_y .

For an $n \times n \times n$ object volume,

$$-n \leq \Delta z_x \leq n,$$

and

$$-n \leq \Delta z_y \leq n.$$

The size of the look-up table will be $(2n + 1) \times (2n + 1)$.

Thus a $16 \times 16 \times 16$ image volume needs a 33×33 look-up table as shown in Table 4.1. The values in Table 4.1 are generated using Equation 4.5 and assuming the light source vector $L = (3, 2, -1)$. This choice is arbitrary so that no redundant values in the look-up table are produced. The phenomenon of redundant values when using a predefined restricted direction of light will be explained later in this section. Once the table is generated, it can be stored in a file for later use. The shading values of a given surface can then be calculated by Equation 3.5 and values of $N(\theta)$ from the look-up table. The look-up table can be applied to other images having the same or lower resolutions. Thus a table generated for the $64 \times 64 \times 64$ array can be used on a $16 \times 16 \times 16$ array image provided the same values of Δz_x and Δz_y are used.

When several look-up tables each associated with a different light source vector are used, an animated sequence of moving light source on a fixed view of an object can be displayed. Such a display technique can be very useful in cartography where 3D shaded images of the shadowing effect of the sun on terrain at different times of the day can be artificially generated. This technique can also be applied on medical images where a particular surface feature of an object can be more prominently seen by a moving light source.

As mentioned previously, but not explained elaborately is the phenomenon of redundant values in the look-up table. When the light source vector is restricted to be lying in the xz -plane or in the yz -plane (as shown in Figs. 4.4a and 4.4b below), the look-up table will have approximately 50% of redundant intensity values.

	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	2	6	9	13	17	22	27	32	38	45	51	58	65	72	78	83	88	92	95	97	98	99	99	98	97	96	95	93	92	91	90	
11	4	7	10	14	17	22	26	31	36	42	48	54	60	66	72	78	83	88	91	94	96	98	98	99	99	98	97	96	95	93	92	91	
12	7	11	14	18	21	26	30	35	40	45	51	56	62	68	73	78	83	87	91	93	96	97	98	99	99	98	97	96	95	93	92	91	
13	11	14	18	21	25	29	33	38	43	48	53	58	64	69	74	79	83	87	90	93	95	96	97	98	98	98	98	98	97	96	95	94	
14	15	18	21	25	28	32	37	41	46	50	55	60	65	70	74	79	83	86	89	92	94	96	97	98	98	98	98	98	97	96	95	94	
15	18	21	24	28	31	35	39	44	48	52	57	62	66	71	75	79	83	86	89	91	93	95	96	97	98	98	98	98	98	97	96	95	
16	21	24	27	30	34	38	42	46	50	54	59	63	67	71	75	79	82	86	88	91	93	94	96	97	97	98	98	98	98	98	97	96	

Table 4.1 A 33 x 33 intensity lookup table for a 16 x 16 x 16 3D image.
 The light source vector is $L = (3, 2, -1)$.
 (Each entry is magnified by a factor of 100).

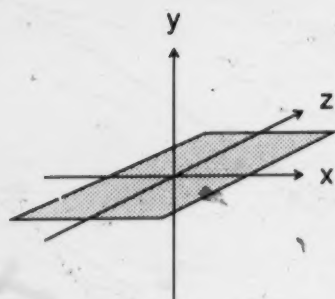


Fig. 4.4a The xz-plane.

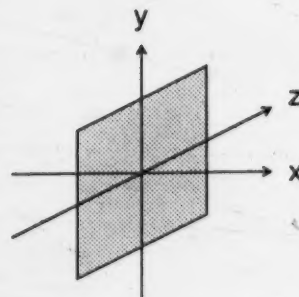


Fig. 4.4b The yz-plane.

If the light source vector is further restricted to be lying along the line of intersection of the xz-plane and the yz-plane (i.e. perpendicular to the viewing screen) then the degree of redundancy will be approximately 87.5%. To explain this, consider first the case when the light source vector is lying on the xz-plane, i.e. $\mathbf{L} = (l_x, 0, l_z)$, substituting this into Equation 4.5, it follows that

$$N(\theta) = \frac{\Delta z_x l_x - 2l_z}{\left\{ \left[(\Delta z_x)^2 + (\Delta z_y)^2 + 4 \right] \cdot \left[l_x^2 + l_z^2 \right] \right\}^{1/2}} \quad (4.6)$$

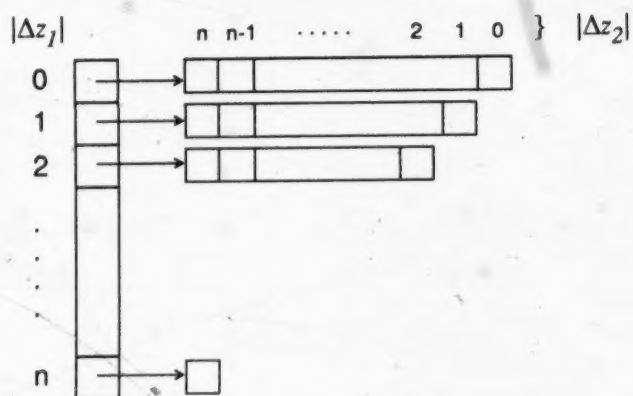
Equation 4.6 shows that altering the sign of Δz_y will not affect the value of $N(\theta)$. This implies that the look-up table will have values on the left half which is a mirror image of the upper half and thus approximately a 50% redundancy. Similar phenomena occur when the light source vector is restricted to be lying on the xy-plane, except that the lower half will be a mirror image of the upper half.

Data redundancy suggests that memory space reduction is possible. By further restricting the light source vector, less memory space is needed to store the look-up table. Certainly, a different data structure other than the 2D array has to be designed to accommodate the space reduction in the look-up table. As mentioned above, by restricting the light source vector to be lying

perpendicular to the viewing screen, a 87.5% reduction approximately is expected. By letting $\mathbf{L} = (0, 0, l_z)$, (here $l_z < 0$, light source vector pointing outwards) and substituting this into Equation 4.5, it follows that

$$N(\theta) = \frac{2}{\left\{ \left[(\Delta z_x)^2 + (\Delta z_y)^2 + 4 \right] \right\}^{1/2}} \quad (4.7)$$

Equation 4.7 shows that altering the sign of Δz_x or Δz_y will not affect the value of $N(\theta)$. This implies that the lower half of the look-up table is a mirror image of the upper half and at the same time the left half is a mirror image of the right half. Careful examination of Equation 4.7 also shows that value of $N(\theta)$ will not be affected even if the values of Δz_x and Δz_y are interchanged. This implies that the upper and lower triangular halves divided by either diagonal are mirror images of each other. This means that the entire look-up table can be sufficiently represented by approximately 1/8th of the table (see Table 4.2). The look-up table then can be stored as a dynamically allocated array as shown in Fig. 4.5. $|\Delta z_1|$ can be the index of the array of pointers (which represents the row index of the look-up table) and $|\Delta z_2|$ can be the index into the dynamically allocated array (which represents the column index of the look-up table).



where $|\Delta z_1| = \max (|\Delta z_x|, |\Delta z_y|)$,
 $|\Delta z_2| = \{ |\Delta z_x|, |\Delta z_y| \} - \{ |\Delta z_1| \}$.

Fig. 4.5 Dynamic array data structure for intensity look-up table.

Algorithms 4.1 and 4.2 depicts the creation of the data structure for the look-up table and the way to look-up the intensity values from this structure. Both algorithms are written as a subroutine in the C language.

	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-16	8	9	9	9	9	10	10	10	11	11	11	11	12	12	12	12	12	12	12	12	12	11	11	11	11	10	10	10	9	9	9	9	8
-15	9	9	9	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13	13	12	12	12	12	11	11	11	11	10	10	10	9	9	9
-14	9	9	10	10	10	11	11	11	12	12	13	13	13	13	14	14	14	14	14	13	13	13	13	12	12	11	11	11	10	10	10	9	9
-13	9	10	10	10	11	11	12	12	12	13	13	14	14	14	15	15	15	15	15	14	14	14	13	13	12	12	12	11	11	10	10	10	9
-12	9	10	10	11	11	12	12	13	13	14	14	15	15	15	16	16	16	16	16	15	15	15	14	14	13	13	12	12	11	11	10	10	9
-11	10	10	11	11	12	12	13	13	14	15	15	16	16	17	17	17	17	17	17	17	16	16	16	15	14	14	13	12	12	11	11	10	10
-10	10	11	11	12	12	13	14	14	15	16	16	17	18	18	19	19	19	19	19	19	18	18	17	16	16	15	14	13	12	12	11	11	10
-9	10	11	11	12	13	13	14	15	16	17	18	19	19	20	21	21	21	21	21	20	19	19	18	17	16	15	14	13	12	12	11	11	10
-8	11	11	12	12	13	14	15	16	17	18	19	20	21	22	23	24	24	24	23	22	21	20	19	18	17	16	15	14	13	12	12	11	11
-7	11	11	12	13	14	15	16	17	18	19	21	22	24	25	26	27	27	27	26	25	24	22	21	19	18	17	16	15	14	13	12	11	11
-6	11	12	13	13	14	15	16	18	19	21	22	24	26	28	30	31	31	31	30	28	26	24	22	21	19	18	16	15	14	13	13	12	11
-5	11	12	13	14	15	16	17	19	20	22	24	27	29	32	34	36	37	36	34	32	29	27	24	22	20	19	17	16	15	14	13	12	11
-4	12	12	13	14	15	16	18	19	21	24	26	29	33	37	40	43	44	43	40	37	33	29	26	24	21	19	18	16	15	14	13	12	12
-3	12	12	13	14	15	17	18	20	22	25	28	32	37	42	48	53	55	53	48	42	37	32	28	25	22	20	18	17	15	14	13	12	12
-2	12	13	14	15	16	17	19	21	23	26	30	34	40	48	57	66	70	66	57	48	40	34	30	26	23	21	19	17	16	15	14	13	12
-1	12	13	14	15	16	17	19	21	24	27	31	36	43	53	66	81	89	81	66	53	43	36	31	27	24	21	19	17	16	15	14	13	12
0	12	13	14	15	16	17	19	21	24	27	31	37	44	55	70	89	100	89	70	55	44	37	31	27	24	21	19	17	16	15	14	13	12
1	12	13	14	15	16	17	19	21	24	27	31	36	43	53	66	81	89	81	66	53	43	36	31	27	24	21	19	17	16	15	14	13	12
2	12	13	14	15	16	17	19	21	23	26	30	34	40	48	57	66	70	66	57	48	40	34	30	26	23	21	19	17	16	15	14	13	12
3	12	12	13	14	15	17	18	20	22	25	28	32	37	42	48	53	55	53	48	42	37	32	28	25	22	20	18	17	15	14	13	12	12
4	12	12	13	14	15	16	18	19	21	24	26	29	33	37	40	43	44	43	40	37	33	29	26	24	21	19	18	16	15	14	13	12	12
5	11	12	13	14	15	16	17	19	20	22	24	27	29	32	34	36	37	36	34	32	29	27	24	22	20	19	17	16	15	14	13	12	11
6	11	12	13	13	14	15	16	18	19	21	22	24	26	28	30	31	31	31	30	28	26	24	22	21	19	18	16	15	14	13	13	12	11
7	11	11	12	13	14	15	16	17	18	19	21	22	24	25	26	27	27	27	26	25	24	22	21	19	18	17	16	15	14	13	12	11	11
8	11	11	12	12	13	14	15	16	17	18	19	20	21	22	23	24	24	24	23	22	21	20	19	18	17	16	15	14	13	12	12	11	11
9	10	11	11	12	13	13	14	15	16	17	18	19	19	20	21	21	21	21	20	19	19	18	17	16	15	14	13	13	12	12	11	11	10
10	10	11	11	12	12	13	14	14	15	16	16	17	18	18	19	19	19	19	18	18	17	16	16	15	14	13	13	12	12	11	11	10	10
11	10	10	11	11	12	12	13	13	14	15	15	16	16	17	17	17	17	17	17	16	16	15	15	14	13	13	12	12	11	11	10	10	10
12	9	10	10	11	11	12	12	13	13	14	14	15	15	15	16	16	16	16	16	15	15	15	14	14	13	13	12	12	11	11	10	10	9
13	9	10	10	10	11	11	12	12	12	13	13	14	14	14	15	15	15	15	15	14	14	14	13	13	12	12	12	11	11	10	10	10	9
14	9	9	10	10	10	11	11	11	12	12	13	13	13	13	14	14	14	14	14	13	13	13	13	12	12	11	11	11	10	10	10	9	9
15	9	9	9	10	10	10	11	11	11	11	12	12	12	12	13	13	13	13	13	12	12	12	12	11	11	11	11	10	10	10	9	9	9
16	8	9	9	9	9	10	10	10	11	11	11	11	12	12	12	12	12	12	12	12	12	11	11	11	11	10	10	10	9	9	9	9	8

Table 4.2 A 33 x 33 intensity lookup table for a 16 x 16 x 16 3D image.
(Each entry is magnified by a factor of 100).

Algorithm 4.1

/* Create dynamic data structure for the look-up table

*** for the light source vector $L = (0, 0, -1)$**

*** N is the size of one side of the voxel volume.**

***/**

static float * l_LUT [N+1];

void Cal_l_LUT (l_LUT)

float * l_LUT [];

{

int i, j;

char * malloc ();

/* 1. Allocate enough space for the LUT */

for (i = 0; i <= N; i++)

l_LUT [i] = (float *) malloc ((N-i+1) * SIZEOF (float));

/* 2. Feed values into the LUT

FUNC (i,j) is the function in Equation 4.7 */

for (i = 0; i <= N; i++)

for (j = i; j <= N; j++)

***(l_LUT [i] + N-j) = FUNC (i, j);**

}

Algorithm 4.2

/* Get Intensity value from the LUT */

float Get_I (I_LUT, delta_z_x, delta_z_y)

float * I_LUT [];

int delta_z_x, delta_z_y; /* -N <= delta_z_x, delta_z_y <= N */

{

int a, b, delta_z₁, delta_z₂;

/* 1. Take the absolute values of both delta_z_x and delta_z_y

so that 0 <= a, b <= N */

a = ABS (delta_z_x);

b = ABS (delta_z_y);

/* 2. Make delta_z₁ the smaller value of {a, b}

and delta_z₂, the bigger value from the set {a, b}.

**This makes sure that the row index always
smaller than the column index of the LUT */**

if (a < b)

{

delta_z₁ = a;

delta_z₂ = b;

}

else

{

delta_z₂ = a;

delta_z₁ = b;

}

**/* 3. Get the value from the LUT and return it to
the calling routine. */**

return (*(I_LUT [delta_z₁] + (N - delta_z₂)))

};

4.3 The Normal-averaging Shading Method

In the display of objects (in CT tomograms) represented by voxels, a difficult dilemma is: on one hand it is required that the object should be displayed with as much detail as possible, preferably by some form of shading that will accentuate the curvature of the surface; on the other hand artifacts, which are a result of the digitizing process, should be eliminated or deemphasized. However, when surface curvatures are emphasized strongly, artifacts stand out sharply. One kind of artifact is produced by the unequal number of photons received on the photo-sensitive elements of the image acquisition devices. If 100 photons were supposed to fall on one pixel, due to the presence of noise (produced by mechanical and electronic parts in the device) there will not be exactly 100 photons on every pixel; some pixels will get less photons and some will get more. This will result in artifacts produced during image segmentation. Another kind of artifact that is closely related to the first one is caused by the quantization of intensity levels (gray-levels). Ordinary industrial imaging devices usually encode in one byte per pixel, i.e. 256 gray-levels. In medical imagery, however, most image acquisition device usually encode in 12 bits per pixel, i.e. an equivalent of 4096 gray-levels. Thus, this is not a significant artifact. The third kind of artifact is produced by the spatial quantization of 3D objects into a discrete voxel representation. The voxel information is then used to generate the shading intensities. In gradient shading each entry of the distance matrix is a measure of the distance (in unit voxel width) of a particular opaque voxel from a specified viewing plane. The surface normal then produced will only be an approximation of the true surface normal. The shaded image generated will possess artifact patterns resulting from a poor approximation of the surface normal. By smoothing the image, artifact patterns could be reduced, but at the risk of eliminating important details. Both methods mentioned in Sections 4.1 and 4.2 produce high quality

images. When displaying objects which have steep slopes or rough surfaces, however, a strong sense of "squareness" resulted. A new gradient shading technique is now proposed that minimizes the artifacts by getting a better approximation of the surface normal and hence produces very high quality images. This method is termed the *normal-averaging method*.

The voxel-based method and the image-space gradient method share a common feature; both methods make use of the four horizontal and vertical neighbors (Fig. 4.6a) for obtaining surface normals. These neighbors are called the *adjacent* neighbors, and the neighbors on the four corners the *diagonal* neighbors (Fig. 4.6b). The normal obtained by either the adjacent neighbors or the diagonal neighbors, however, does not always represent the actual surface orientation. For instance, Figs. 4.7a, 4.7b, and 4.7c show one example out of the many situations where the actual surface normal is poorly estimated by either method alone. A new technique is therefore proposed for getting a better estimation of the the surface normal.

The method adopted is to estimate the actual surface normal by means of :

- (1) $\mathbf{n}_a$, the unit normal (i.e. normalized $\mathbf{N}_a$) obtained by means of the adjacent neighbors only, and
- (2) $\mathbf{n}_d$, the unit normal (i.e. normalized $\mathbf{N}_d$) obtained by means of the diagonal neighbors only, and
- (3) two weighting factors w_a and w_d .

As shown in Fig. 4.7d, the actual normal $\mathbf{N}$ can be obtained by the Equation:

$$\mathbf{N} = w_a \cdot \mathbf{n}_a + w_d \cdot \mathbf{n}_d. \quad (4.8)$$

The two weighting factors are empirical parameters for adjusting the direction of $\mathbf{N}$. If $w_a = w_d = 1$, then $\mathbf{N}$ will be equally "tilted" between $\mathbf{n}_a$ and $\mathbf{n}_d$. If the

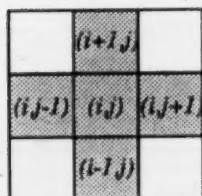


Fig. 4.6a A 3 x 3 portion of the distance matrix. Shaded parts are the centre and its adjacent neighbors.

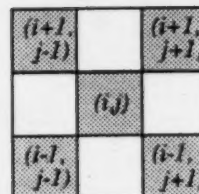


Fig. 4.6b A 3 x 3 portion of the distance matrix. Shaded parts are the centre and its diagonal neighbors.

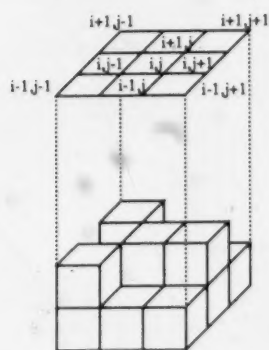


Fig. 4.7a A 3 x 3 portion of the distance matrix and the corresponding object surface.

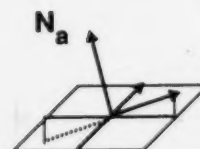


Fig. 4.7b The normal based on adjacent neighbors.

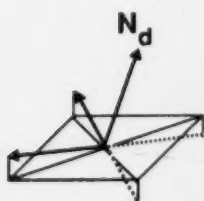


Fig. 4.7c The normal based on diagonal neighbors.

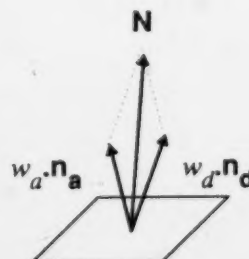


Fig. 4.7d The averaged normal.

closeness of the neighboring elements is taken into account with respect to the central element, then using the fact that adjacent neighbors are 1 unit away and the diagonal neighbors are $\sqrt{2}$ units away, we can assign $w_a=1$ and $w_d=1/\sqrt{2}$. To obtain the surface normal of adjacent neighbors, N_a , use Equations 4.2, 4.3 and the following backward, forward, and central differences for $\partial z/\partial x$:

backward difference

$$\delta_b = z_{ij} - z_{i-1,j}, \quad (2 \leq i \leq M, 1 \leq j \leq M), \quad (4.9a)$$

forward difference

$$\delta_f = z_{i+1,j} - z_{ij}, \quad (1 \leq i \leq M-1, 1 \leq j \leq M), \quad (4.9b)$$

central difference

$$\delta_c = \frac{1}{2} (z_{i+1,j} - z_{i-1,j}), \quad (2 \leq i \leq M-1, 1 \leq j \leq M). \quad (4.9c)$$

Similar techniques can be used to obtain $\partial z/\partial y$. For the surface normal of diagonal neighbors, N_d , use Equations 4.2, 4.3 and the following backward, forward and central differences for $\partial z/\partial x'$ (define X' to be the diagonal axis in the SW-NE direction and Y' to be the diagonal axis in the SE-NW direction):

backward difference

$$\delta_b = z_{ij} - z_{i-1,j-1}, \quad (2 \leq i \leq M, 2 \leq j \leq M), \quad (4.10a)$$

forward difference

$$\delta_f = z_{i+1,j+1} - z_{ij}, \quad (1 \leq i \leq M-1, 1 \leq j \leq M-1), \quad (4.10b)$$

central difference

$$\delta_c = \frac{1}{2} (z_{i+1,j+1} - z_{i-1,j-1}), \quad (2 \leq i \leq M-1, 2 \leq j \leq M-1). \quad (4.10c)$$

And these differences for $\partial z / \partial y'$:

backward difference

$$\delta_b = z_{ij} - z_{i-1,j+1}, \quad (2 \leq i \leq M, 2 \leq j \leq M), \quad (4.11a)$$

forward difference

$$\delta_f = z_{i+1,j-1} - z_{ij}, \quad (1 \leq i \leq M-1, 1 \leq j \leq M-1), \quad (4.11b)$$

central difference

$$\delta_c = \frac{1}{2} (z_{i+1,j-1} - z_{i-1,j+1}), \quad (2 \leq i \leq M-1, 2 \leq j \leq M-1). \quad (4.11c)$$

By letting $N(\theta) = (\cos \theta)^p = (\mathbf{N} \cdot \mathbf{L} / (|\mathbf{N}| \cdot |\mathbf{L}|))^p$ and using Equation 3.5, the shading intensity for each point of the object can be obtained. The normal-averaging method does give better quality images than the previous two methods. In order to produce even smoother images, reduce the angular dependency on the cosine function by keeping $p < 1$. Experiments show that $p=0.6$ produces high quality images.

4.5 Qualitative analysis of different gradient shading methods

For gradient shading, the key to better image quality lies in the estimation of the surface normal. The closer the estimated normal to the actual surface normal, the better the shaded image quality. To compare the image quality of different gradient shading methods discussed above, it is necessary to compare the closeness of the surface normals of different methods with the actual surface normal. Three types of gradient shading methods will be compared in this section:

- (1) the gradient shading using adjacent neighbors of the distance matrix (image-space gradient shading method by Gordon and Reynolds [17]),
- (2) the gradient shading using diagonal neighbors of the distance matrix, and
- (3) the gradient shading using both the adjacent neighbors and the diagonal neighbors of the distance matrix (the normal-averaging shading method developed in this thesis).

First, a function has to be picked for generating the object surface and hence the actual surface normal and the distance matrix. The function of a sphere is ideal for comparison because no two different points on the surface of the sphere will produce the same surface normal and thus give different surface normals for comparison at different points. The following function of a sphere is employed,

$$f(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0, \quad r^2 \text{ is constant.}$$

The actual surface normal of function f can be obtained by the gradient of f ,

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}.$$

Here

$$\nabla f = \mathbf{N}_T = 2x \mathbf{i} + 2y \mathbf{j} + 2z \mathbf{k}$$

and

$$\mathbf{n}_T = \frac{\mathbf{N}_T}{|\mathbf{N}_T|} = \frac{(x, y, z)}{(x^2 + y^2 + z^2)^{1/2}} = \frac{(x, y, z)}{r} \quad (4.12)$$

where $\mathbf{N}_T$ is the actual normal vector and $\mathbf{n}_T$ is the unit vector. Let $\mathbf{n}_{adj}$, $\mathbf{n}_{diag}$, and $\mathbf{n}_{avg}$ be the unit vectors for the normals generated by adjacent neighbors (image-space gradient shading), diagonal neighbors, and the combined adjacent and diagonal neighbors (normal-averaging shading) of the distance matrix. To compare the accuracy of each unit surface normal, the dot product function is employed. The dot product of two unit vectors equals 1 if both vectors are colinear and in the same direction and get smaller when they are farther and farther apart until the product becomes -1 which indicates the two vectors are colinear and in opposite directions. By comparing the values of the three dot products $\mathbf{n}_T \cdot \mathbf{n}_{adj}$, $\mathbf{n}_T \cdot \mathbf{n}_{diag}$, and $\mathbf{n}_T \cdot \mathbf{n}_{avg}$, one would be able to compare the image quality; the higher the dot product, the better the image quality. Table 4.3 shows the distance matrix of the digital version of the sphere: $x^2 + y^2 + z^2 = 16^2$. The values in the distance matrix simulates the image acquisition process in medical imagery where a 3D object is represented by discrete voxels. Tables 4.4a, 4.4b, 4.4c show the values obtained from the dot products $\mathbf{n}_T \cdot \mathbf{n}_{adj}$, $\mathbf{n}_T \cdot \mathbf{n}_{diag}$, and $\mathbf{n}_T \cdot \mathbf{n}_{avg}$ respectively. $\mathbf{n}_{adj}$ is obtained from Equations 4.1, 4.2 and 4.3. $\mathbf{n}_{diag}$ is obtained from Equations 4.2, 4.3 and 4.10. $\mathbf{n}_{avg}$ is obtained from Equations 4.2, 4.3, 4.8, 4.9, and 4.10 (and 4.11). The bold values in Tables 4.4a, 4.4b, 4.4c indicates dot products of less than 0.96 (an arbitrary choice for comparison) which translates into about 16° or more angle difference between the unit vectors. Table 4.4a shows a cluster of bold values near the centre of the table and some smaller clusters that radiates out from the centre. The lowest value in this table is 88 (which implies a dot product of 0.88 or 28° angular difference). The cluster of bold values near the centre implies that the *image-space gradient shading method* will perform poorly on

surfaces with small values of $\partial z/\partial x$ and $\partial z/\partial y$. This is a very significant drawback, because when this shading method is applied to smooth surfaces (or surfaces with small $\partial z/\partial x$ and $\partial z/\partial y$) artifact patterns are bound to appear. Table 4.4b shows a different picture where the bold values are evenly distributed all over the sphere. This implies that the gradient shading using diagonal neighbors will produce smoother images in appearance, but poorer approximation to the real object surface on the average case. When compared with the adjacent neighbors, this shading method will produce images with more consistent quality. Table 4.4c shows a much cleaner picture with fewer bold values and the lowest among them is 92 (dot product of 0.92 or 23° angular difference). Due to the added effect of the diagonal neighbors, the central clustering of bold values has been reduced and de-centralized. This means that the *normal-averaging shading method* can perform better on smooth surfaces and give a better approximation to the actual surface normal and hence produces better quality images! Thus the *normal-averaging shading method* is superior than the *image-space gradient shading method*!

	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-16	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
-15	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	4	5	5	5	6	5	5	5	4	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
-14	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	5	6	7	7	7	8	8	8	7	7	7	6	5	3	-1	-1	-1	-1	-1	-1	-1	-1	
-13	-1	-1	-1	-1	-1	-1	-1	2	5	6	7	8	8	9	9	9	9	9	9	9	8	8	7	6	5	2	-1	-1	-1	-1	-1	-1	
-12	-1	-1	-1	-1	-1	-1	3	6	7	8	9	9	10	10	10	11	11	11	10	10	10	9	9	8	7	6	3	-1	-1	-1	-1	-1	
-11	-1	-1	-1	-1	-1	4	6	7	8	9	10	10	11	11	11	12	12	12	11	11	11	10	10	9	8	7	6	4	-1	-1	-1	-1	
-10	-1	-1	-1	-1	3	6	7	9	10	10	11	11	12	12	12	12	12	12	12	12	12	11	11	10	10	9	7	6	3	-1	-1	-1	
-9	-1	-1	-1	2	6	7	9	10	11	11	12	12	13	13	13	13	13	13	13	13	13	12	12	11	11	10	9	7	6	2	-1	-1	
-8	-1	-1	-1	5	7	8	10	11	11	12	12	13	13	14	14	14	14	14	14	14	14	13	12	12	11	11	10	8	7	5	-1	-1	
-7	-1	-1	3	6	8	9	10	11	12	13	13	13	14	14	14	14	14	14	14	14	14	13	13	13	12	11	10	9	8	6	3	-1	
-6	-1	-1	5	7	9	10	11	12	12	13	14	14	14	15	15	15	15	15	15	15	14	14	14	13	12	12	11	10	9	7	5	-1	
-5	-1	2	6	8	9	10	11	12	13	13	14	14	15	15	15	15	15	15	15	15	15	14	14	13	13	12	11	10	9	8	6	2	
-4	-1	4	7	8	10	11	12	13	13	14	14	15	15	15	15	15	15	15	15	15	15	15	14	14	13	13	12	11	10	8	7	4	
-3	-1	5	7	9	10	11	12	13	14	14	15	15	15	15	16	16	16	16	16	16	15	15	15	15	14	14	13	12	11	10	9	7	
-2	-1	5	7	9	10	11	12	13	14	14	15	15	15	16	16	16	16	16	16	16	16	15	15	15	14	14	13	12	11	10	9	7	
-1	-1	5	8	9	11	12	12	13	14	14	15	15	15	16	16	16	16	16	16	16	16	15	15	15	14	14	13	12	12	11	9	8	
X 0	0	6	8	9	11	12	12	13	14	14	15	15	15	16	16	16	16	16	16	16	16	15	15	15	14	14	13	12	12	11	9	8	
1	-1	5	8	9	11	12	12	13	14	14	15	15	15	16	16	16	16	16	16	16	16	15	15	15	14	14	13	12	12	11	9	8	
2	-1	5	7	9	10	11	12	13	14	14	15	15	15	16	16	16	16	16	16	16	16	15	15	15	14	14	13	12	11	10	9	7	
3	-1	5	7	9	10	11	12	13	14	14	15	15	15	16	16	16	16	16	16	16	15	15	15	15	14	14	13	12	11	10	9	7	
4	-1	4	7	8	10	11	12	13	13	14	14	15	15	15	15	15	15	15	15	15	15	15	14	14	13	13	12	11	10	8	7		
5	-1	2	6	8	9	10	11	12	13	13	14	14	15	15	15	15	15	15	15	15	15	14	14	13	13	12	11	10	9	8	6		
6	-1	-1	5	7	9	10	11	12	12	13	14	14	14	15	15	15	15	15	15	15	14	14	14	13	12	12	11	10	9	7	5		
7	-1	-1	3	6	8	9	10	11	12	13	13	13	14	14	14	14	14	14	14	14	14	13	13	13	12	11	10	9	8	6	3		
8	-1	-1	-1	5	7	8	10	11	11	12	12	13	13	14	14	14	14	14	14	14	14	13	13	12	12	11	11	10	8	7	5		
9	-1	-1	-1	2	6	7	9	10	11	11	12	12	13	13	13	13	13	13	13	13	13	12	12	11	11	10	9	7	6	2			
10	-1	-1	-1	-1	3	6	7	9	10	10	11	11	12	12	12	12	12	12	12	12	12	11	11	10	10	9	7	6	3	-1			
11	-1	-1	-1	-1	-1	4	6	7	8	9	10	10	11	11	11	12	12	12	11	11	11	10	10	9	8	7	6	4	-1	-1			
12	-1	-1	-1	-1	-1	-1	3	6	7	8	9	9	10	10	10	11	11	11	10	10	10	10	9	9	8	7	6	3	-1	-1			
13	-1	-1	-1	-1	-1	-1	-1	2	5	6	7	8	8	9	9	9	9	9	9	9	9	8	8	7	6	5	2	-1	-1				
14	-1	-1	-1	-1	-1	-1	-1	-1	3	5	6	7	7	7	8	8	8	7	7	7	7	6	5	3	-1	-1	-1	-1	-1				
15	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	4	5	5	5	6	5	5	5	5	4	2	-1	-1	-1	-1	-1	-1	-1				
16	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1				

Table 4.3 A distance matrix of the digital sphere: $x^2 + y^2 + z^2 = 16^2$.
(-1 indicates invalid distance values since no object is found).

Y

	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
-16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	98	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
-15	X	X	X	X	X	X	X	X	X	X	X	98	97	98	98	99	99	99	98	98	97	98	X	X	X	X	X	X	X	X	X	X	X	
-14	X	X	X	X	X	X	X	X	X	99	97	99	99	98	99	98	99	98	99	98	99	99	97	99	X	X	X	X	X	X	X	X	X	
-13	X	X	X	X	X	X	99	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	98	99	X	X	X	X	X	X	X	
-12	X	X	X	X	X	99	97	99	99	99	99	99	97	97	97	97	97	97	97	97	97	99	99	99	99	97	99	X	X	X	X	X	X	
-11	X	X	X	X	X	99	99	99	99	98	99	99	99	98	97	89	95	89	97	98	99	99	99	99	99	99	99	99	X	X	X	X	X	
-10	X	X	X	X	99	99	99	99	95	99	99	99	99	97	98	97	97	97	98	97	99	99	99	95	99	99	99	99	X	X	X	X	X	
-9	X	X	X	99	97	99	99	99	97	98	98	98	97	96	97	98	98	98	97	96	97	98	98	98	97	99	99	99	97	99	X	X	X	
-8	X	X	X	98	99	99	95	97	98	97	99	99	98	97	99	99	99	99	99	99	97	98	99	99	97	98	97	95	99	99	98	X	X	X
-7	X	X	99	99	99	98	99	98	97	99	88	99	98	98	99	99	99	99	99	98	98	99	88	99	97	98	99	98	99	99	99	X	X	X
-6	X	X	97	99	99	99	99	98	99	88	99	94	98	97	98	99	99	99	98	97	98	94	99	88	99	98	99	99	99	99	97	X	X	X
-5	X	98	99	99	99	99	99	98	99	99	94	98	97	93	94	94	94	94	94	93	97	98	94	99	99	98	99	99	99	99	99	98	X	X
-4	X	97	99	99	99	99	99	97	98	98	98	97	93	94	97	97	97	97	97	94	93	97	98	98	98	97	99	99	99	99	99	97	X	X
-3	X	98	98	99	97	98	97	96	97	98	97	93	94	94	92	96	96	96	92	94	94	93	97	98	97	96	97	98	97	99	98	98	X	X
-2	X	98	99	99	97	97	98	97	99	99	98	94	97	92	98	99	99	99	98	92	97	94	98	99	99	97	98	97	97	99	99	98	X	X
-1	X	99	98	99	97	89	97	98	99	99	99	94	97	96	99	99	99	99	99	96	97	94	99	99	99	98	97	89	97	99	98	99	X	X
X 0	98	99	99	99	99	95	97	98	99	99	99	94	97	96	99	99	100	99	99	96	97	94	99	99	99	98	97	95	99	99	99	99	98	X
1	X	99	98	99	97	89	97	98	99	99	99	94	97	96	99	99	99	99	99	96	97	94	99	99	99	98	97	89	97	99	98	99	X	X
2	X	98	99	99	97	97	98	97	99	99	98	94	97	92	98	99	99	99	98	92	97	94	98	99	99	97	98	97	97	99	99	98	X	X
3	X	98	98	99	97	98	97	96	97	98	97	93	94	94	92	96	96	96	92	94	94	93	97	98	97	96	97	98	97	99	98	98	X	X
4	X	97	99	99	99	99	99	97	98	98	98	97	93	94	97	97	97	97	97	94	93	97	98	98	98	97	99	99	99	99	99	97	X	X
5	X	98	99	99	99	99	99	98	99	99	94	98	97	93	94	94	94	94	94	93	97	98	94	99	99	98	99	99	99	99	99	98	X	X
6	X	X	97	99	99	99	99	98	99	88	99	94	98	97	98	99	99	99	98	97	98	94	99	88	99	98	99	99	99	99	97	X	X	X
7	X	X	99	99	99	98	99	98	97	99	88	99	98	98	99	99	99	99	99	98	98	99	88	99	97	98	99	98	99	99	99	X	X	X
8	X	X	X	98	99	99	95	97	98	97	99	99	98	97	99	99	99	99	99	97	98	99	99	97	98	97	95	99	99	98	X	X	X	X
9	X	X	X	99	97	99	99	99	97	98	98	98	97	96	97	98	98	98	97	96	97	98	98	98	97	99	99	99	97	99	X	X	X	X
10	X	X	X	X	99	99	99	99	95	99	99	99	99	99	99	99	99	99	99	97	98	99	99	99	95	99	99	99	99	X	X	X	X	X
11	X	X	X	X	X	99	99	99	99	99	99	99	99	99	98	97	89	95	89	97	98	99	99	99	98	99	99	99	99	X	X	X	X	X
12	X	X	X	X	X	X	99	97	99	99	99	99	99	99	97	97	97	97	97	97	99	99	99	99	99	97	99	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	99	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	98	99	X	X	X	X	X	X	X
14	X	X	X	X	X	X	X	X	99	97	99	99	98	99	98	99	98	99	98	99	99	99	99	99	99	99	X	X	X	X	X	X	X	X
15	X	X	X	X	X	X	X	X	X	X	X	98	97	98	98	99	99	99	98	98	97	98	X	X	X	X	X	X	X	X	X	X	X	X
16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	98	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 4.4a Dot products of $\mathbf{n}_T \cdot \mathbf{n}_{adj}$. Bold values are values under 96 which implies $\mathbf{n}_{adj}$ is farther away from $\mathbf{n}_T$. (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).

X

Table 4.4b Dot products of $\mathbf{n_T} \cdot \mathbf{n_{diag}}$. Bold values are values under 96 which implies $\mathbf{n_{diag}}$ is farther away from $\mathbf{n_T}$. (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).

X

Table 4.4c Dot products of $\mathbf{n}_T \cdot \mathbf{n}_{avg}$. Bold values are values under 96 which implies $\mathbf{n}_{avg}$ is farther away from $\mathbf{n}_T$. (Each entry is magnified by a factor of 100). (X indicates invalid dot product values since no object is found).

4.5 Experimental Results

Two objects are used to illustrate the images produced by the gradient shading methods described in this chapter. The first object is a digital approximation of a sphere. The second data set is obtained from SPECT (Single Photon Emission Computed Tomography) imaging of a liver. The SPECT images are obtained from the General Electric 400 A/T gamma camera. The diameter of the sphere is 64 voxels, and the maximum size of the box that encloses the liver is 64 voxels. Each voxel has the same size in all three dimensions. The SPECT data of the liver is preprocessed by applying a global thresholding segmentation technique described in Chapter 2. In Figs. 4.8, 4.9, 4.10, and 4.11, the images shown are produced by different gradient shading techniques (for the two data sets and variations in orientations of the liver data) in the order:

- (a) voxel-based shading method,
- (b) image-space gradient shading method with diagonal neighbors and $p=1$,
- (c) image-space gradient shading method with adjacent neighbors and $p=1$,
- (d) normal-averaging method with $p=1$,
- (e) image-space gradient shading method with adjacent neighbors and $p=0.6$,
and
- (f) normal-averaging method with $p=0.6$.

The orientation of the liver in Fig. 4.10 is at a 30° tilt upwards in the vertical plane. The liver in Fig. 4.11 is generated through a horizontal swing of 15° (patient's right to patient's left). To demonstrate the flexibility of gradient shading, exposed views of the liver are also provided. The exposed views are generated from a coronal image with "cuts" at distances of 20, 25, 30, and 35 pixels from the first image plane at which the light source is assumed to be located.

The voxel-based shading method tends to emphasize artifacts due to the

poor performance on smooth surfaces (as explained in the previous section on shading using adjacent neighbors). This is apparent in Figs. 4.8a, 4.9a, 4.10a, and 4.11a. As discussed earlier, the voxel-based shading method is particularly poor in approximating surface normals at discontinuities, this can be seen in Fig. 4.10a where there is a steep slope between the upper part and the lower part of the right liver half (patient's right liver half, i.e. left half shown in the photo).

The image-space gradient shading method reduces the artifacts due to surface discontinuities by using a weighting function (4.2 and 4.3), the result of which is smoother images at discontinuities. This can be seen in Fig. 4.10c. Two kinds of images based on the image-space gradient shading method are produced: one based on diagonal neighbors (Figs. 4.8b, 4.9b, 4.10b, and 4.11b) and the other based on adjacent neighbors (Figs. 4.8c, 4.9c, 4.10c, and 4.11c). The diagonal-neighbor method tends to produce smoother images than the adjacent-neighbor method (again explained in the previous section), however, many details of the image are lost. The image-space gradient shading method, in general, produces better images than the voxel-based shading method, yet artifacts are still quite noticeable.

The newly developed normal-averaging method seems to produce the best quality images among all the gradient shading methods discussed in this thesis. The results can be seen in Figs. 4.8d, 4.9d, 4.10d, and 4.11d.

To illustrate the effect of angular dependency on the cosine function in $N(\theta) = (\cos \theta)^p$, in Figs. 4.8e, 4.8f, 4.9e, 4.9f, 4.10e, 4.10f, 4.11e, and 4.11f, the effect of the parameter p on images is shown. For all these figures the images are generated with $p=0.6$. All of the Figs. 4.xe are produced by the image-space gradient shading method using adjacent neighbors, and all Figs. 4.xf

are produced by the normal-averaging shading method. By comparing all Figs. 4.xe with Figs. 4.xc and all Figs. 4.xf with 4.xd, it can be seen that images with $p=0.6$ give smoother shades than images with $p=1$. In some cases, more detail can be seen at surfaces with steep slope. In other cases, object surface details are not so apparent on the smoother images with $p=0.6$. On the basis of this dilemma, it is suggested that parameter p could be used as a software "knob" for fine tuning the images to the desired quality.

To demonstrate further the capability of the gradient shading method for producing high quality exposed views of objects, four exposed views of the liver with "cuts" at 20, 25, 30, and 35 voxels from the first image plane are also included in Fig. 4.12. The hollow space inside the liver reflects a tumor being exaggerated by a global thresholding segmentation on the SPECT images. (Here a very large threshold is used to exaggerate the hollow space inside the liver for showing the effect of shading). In the case of surgical planning, seeing the exposed views of internal organs are invaluable to clinicians.

The shaded images are generated on an Apollo DN560 workstation using C as the programming language. It took an average of 80 seconds to produce a distance matrix from a $64 \times 64 \times 64$ voxel image. It took 0.5 seconds to produce a shaded image using the voxel-based shading method, 1 second for the image-space gradient shading method, and 1.5 second for the normal-averaging shading method. In general, it takes far less time to produce the shaded image than to find the distance matrix. An interesting topic for further research is to implement the distance matrix and gradient shading techniques on special hardware or parallel processors. This provides the opportunity for displaying the image in near-real time or even real-time.

National Library
of Canada

Canadian Theses Service

Bibliothèque nationale
du Canada

Service des thèses canadiennes

NOTICE

THE QUALITY OF THIS MICROFICHE
IS HEAVILY DEPENDENT UPON THE
QUALITY OF THE THESIS SUBMITTED
FOR MICROFILMING.

UNFORTUNATELY THE COLOURED
ILLUSTRATIONS OF THIS THESIS
CAN ONLY YIELD DIFFERENT TONES
OF GREY.

AVIS

LA QUALITE DE CETTE MICROFICHE
DEPEND GRANDEMENT DE LA QUALITE DE LA
THESE SOUMISE AU MICROFILMAGE.

MALHEUREUSEMENT, LES DIFFERENTES
ILLUSTRATIONS EN COULEURS DE CETTE
THESE NE PEUVENT DONNER QUE DES
TEINTES DE GRIS.

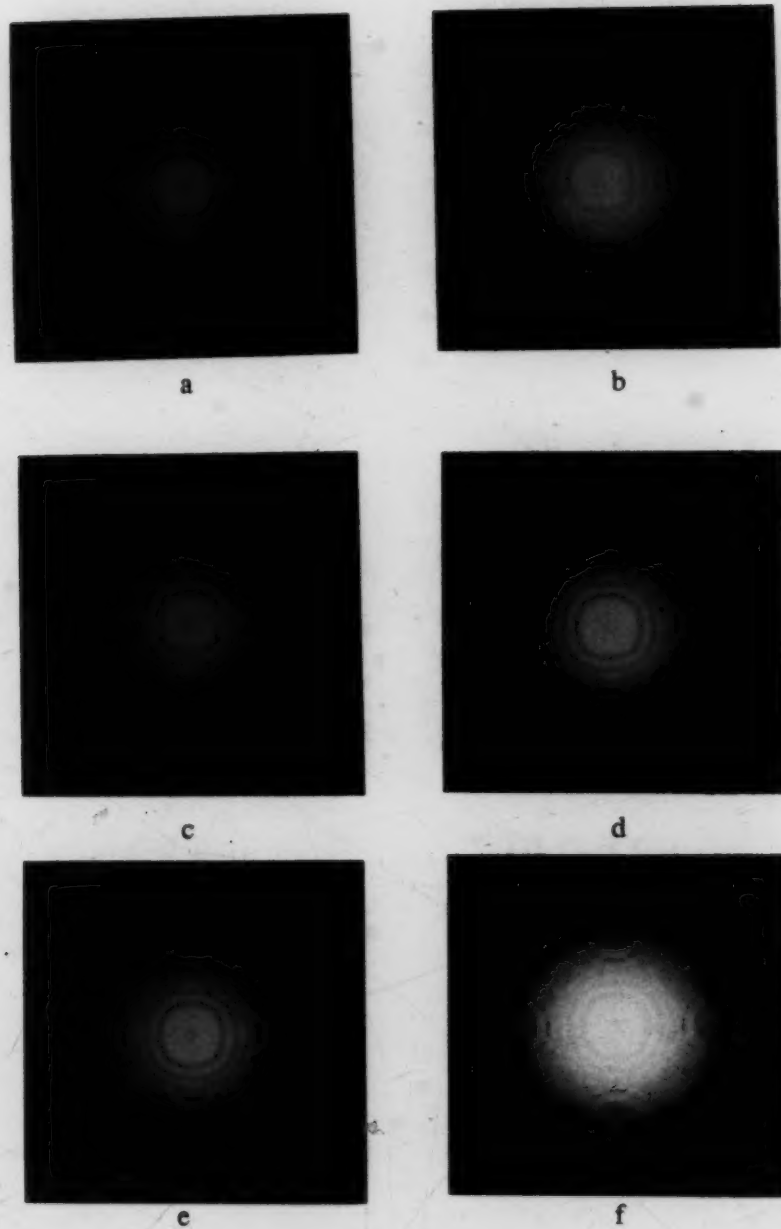


Fig. 4.8 3D display of a digital sphere.

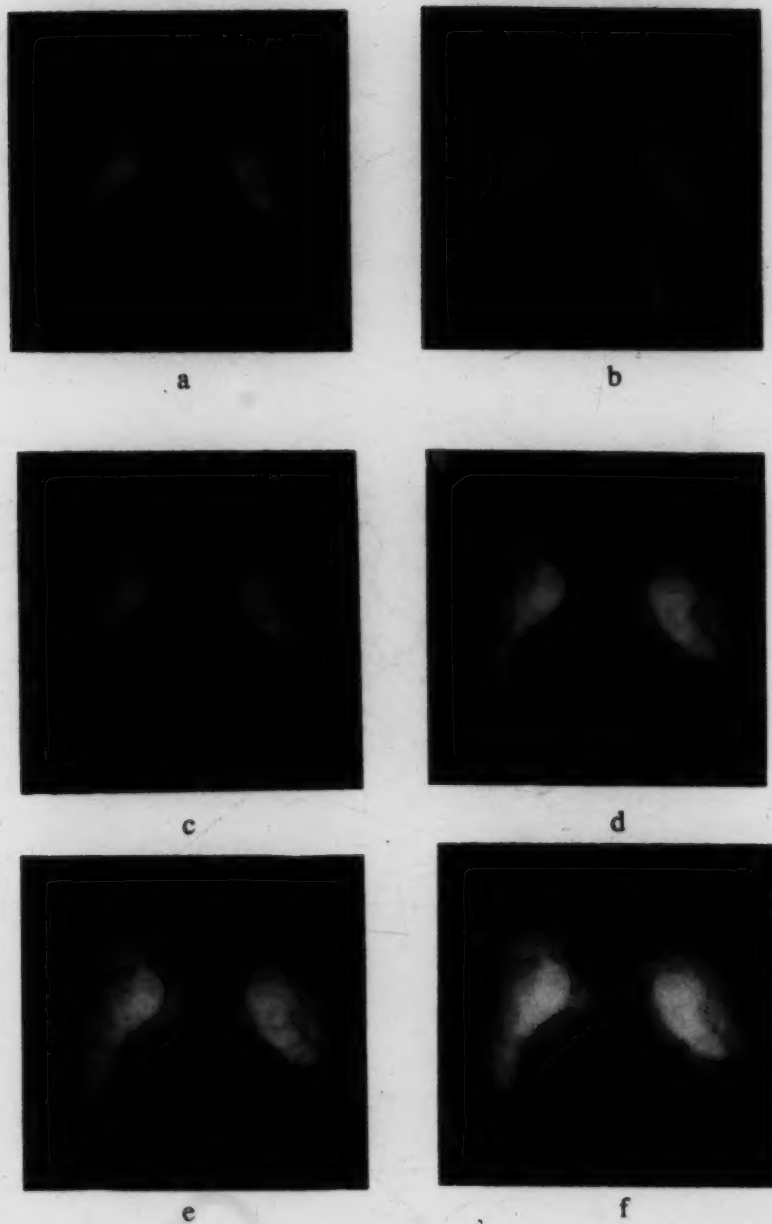


Fig. 4.9 3D display of a liver (coronal view).

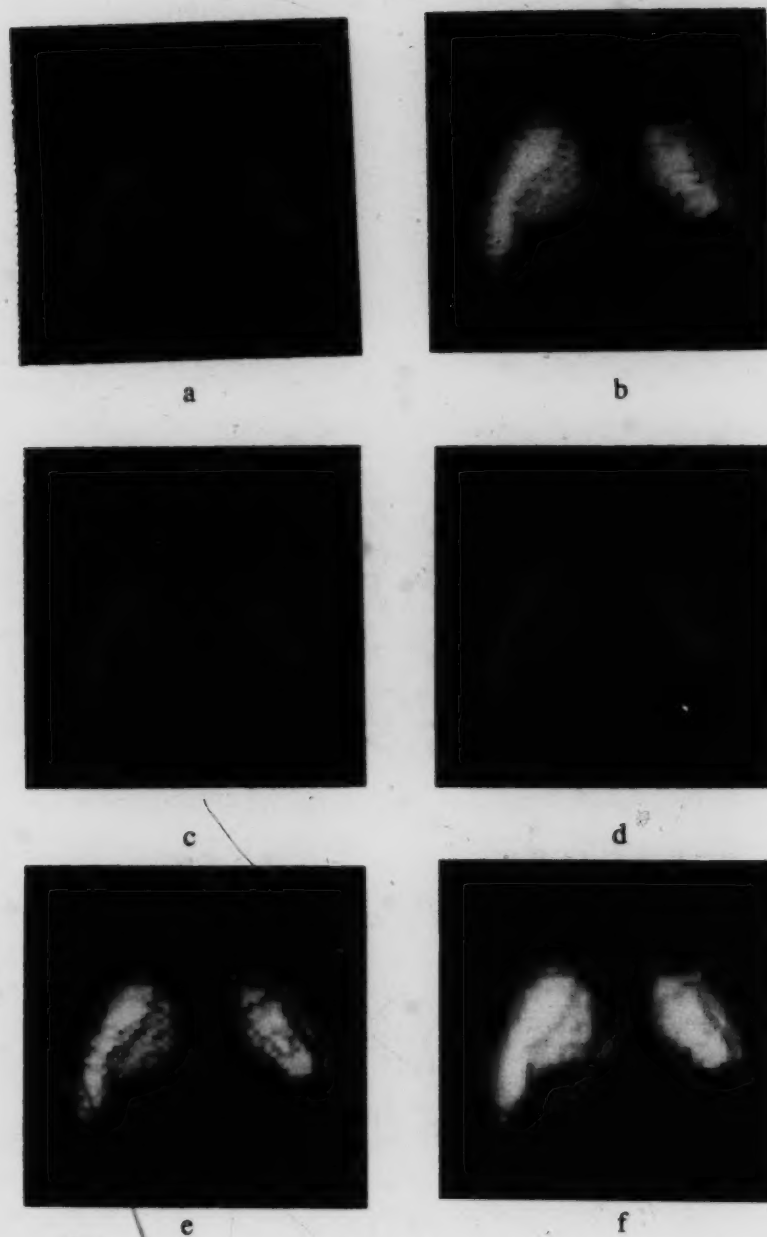


Fig. 4.10 3D display of a liver (coronal view tilted 30° upwards in the vertical plane).

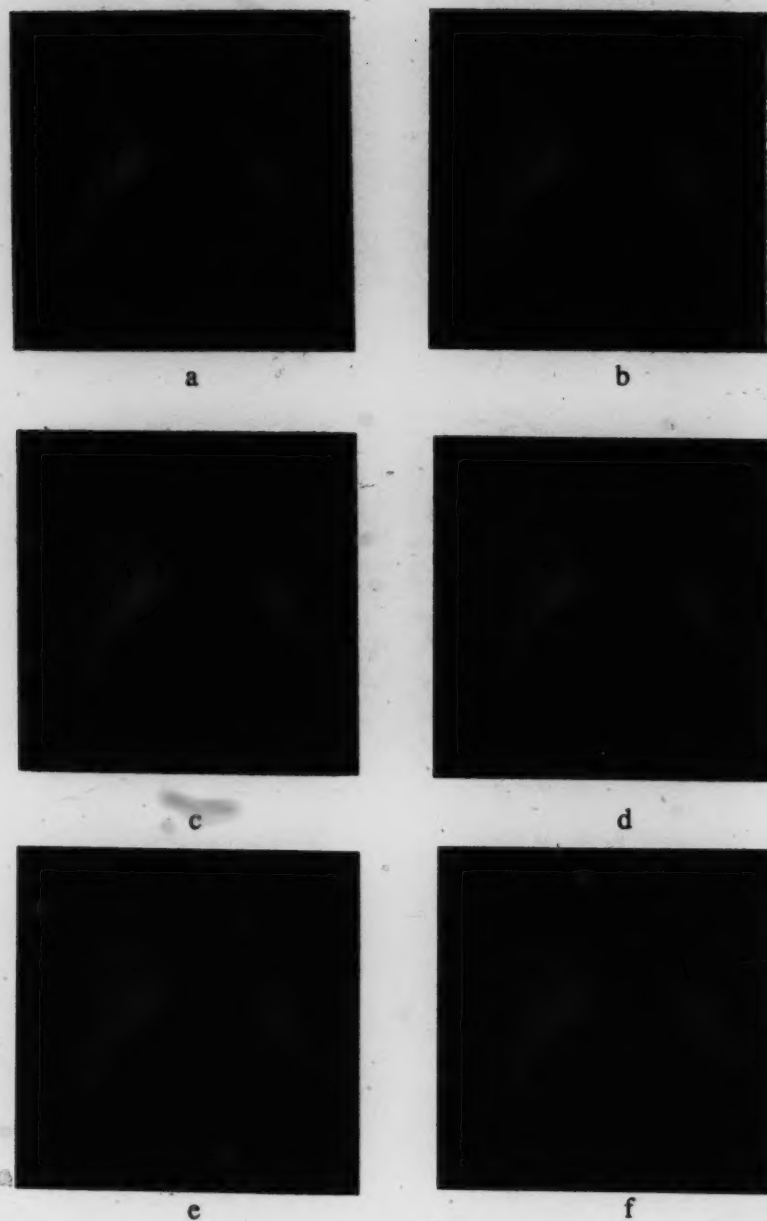


Fig. 4.11 3D display of a liver (coronal view rotated 15° from patient's right to left).



Fig. 4.12 3D display of four exposed view of a liver (with cuts at 20, 25, 30, and 35 voxels respectively from the first image plane).

Chapter 5 - CONCLUSIONS

This thesis deals with the display (especially using shading) of 3D data sets with emphasis in medical imaging applications. The following steps for displaying 3D image data have been described and discussed:

- (1) image segmentation,
- (2) merging of 2D data into a 3D representation, and
- (3) 3D object display.

Global thresholding has been found, by far, the most appropriate and most widely used form of image segmentation applied to 3D medical imaging [15, 16, 21, 27]. In particular in the applications such as surgical planning, thresholding has been found to be an effective means of image segmentation before shading is applied. Thresholding also allows a significant saving of memory space in object representation since binary 0's and 1's are sufficient to represent the segmented object.

3D data representation techniques such as triangular tiles, splines, octree, cuberille and voxels have been described. Recent research interests in 3D representation and shading [4, 5, 15, 18, 21-25, 27] focused mainly on the cuberille and voxel representation models. The cuberille and the voxel models essentially represent the same thing -- a cubic volume element in 3D space. They differ mainly in the application of shading techniques in each of the models. In the cuberille model, shading is determined by the orientation of the *faces* of the volume element and neighboring cuberilles. In the voxel model, shading is determined by the location of the *centre* of the volume element and the voxels.

A number of shading techniques based on the cuberille model of

representation have been discussed in Chapter 3 of this thesis. Chapter 4 discussed, in details, shading techniques based on the voxel model of representation. This type of technique is called a gradient shading technique.

Three gradient shading methods have been investigated:

- (1) the image-space gradient shading method proposed by Gordon and Reynolds [18],
- (2) the voxel-based gradient shading method developed at an earlier stage of this research, and
- (3) the normal-averaging gradient shading method which is newly developed.

The normal-averaging gradient shading method is found to produce the best quality images among the three methods and its better performance is based on a better approximation of the surface normal. The qualitative analysis performed in Section 4.4 explained why this method is better and Section 4.5 verified the results.

Gradient shading methods, in general, produce high quality images more efficiently than many other shading methods and it is a tool that is indispensable in the area of interactive surgical planning. One prominent feature of gradient shading is its efficiency in providing exposed views of objects that other shading techniques find difficult to match. For the liver images in Figs. 4.9 - 4.12, the average time for obtaining a depth matrix is about 80 seconds and the average time to generate a shaded image (using the *Normal-averaging method*) is 1.5 second. This suggests that if parallel processors were used to compute the distance matrix, a significant speed up could be attained.

REFERENCES

- [1] E. Artzy, G. Frieder, and G.T. Herman, "The Theory, Design, Implementation and Evaluation of a Three-Dimensional Surface Detection Algorithm", *Computer Graphics and Image Processing*, Vol. 1, 1981, pp. 1-24.
- [2] D.N. Ballard and C.M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, New Jersey, 1982.
- [3] J.F. Canny, "Finding Edges and Lines in Images", *MIT Artificial Intelligence Lab*, Technical Report No. 720, June 1983.
- [4] L. Chen, G.T. Herman, R.A. Reynolds, and J.K. Udupa, "Surface Shading in the Cuberille Environment", *IEEE Computer Graphics and Applications*, Vol. 5, Dec. 1985, pp. 33-43.
- [5] T.Y. Chen S. Pitluck, and T. Richards, "Three-Dimensional Graphics in Radiation Therapy", *Computer Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp. 131-147.
- [6] C.K. Chow and T. Kaneko, "Automatic Detection of the Left Ventricle From Cineangiograms", *Computers and Biomedical Research*, Vol. 5, Aug. 1972, pp. 388-410.
- [7] L.T. Cook, S.J. Dwyer III, S. Batnitzky, and K.R. Lee, "A Three-Dimensional Display System for Diagnostic Imaging Applications", *IEEE Computer Graphics and Applications*, Vol. 3, Aug. 1983, pp. 13-46.
- [8] H.N. Christiansen and T.W. Sederberg, "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics", *Computer Graphics*, Vol. 12, Aug. 1978, pp. 187-192.
- [9] W.A. Davis and S.R. Venkatramanan, "Image Operations to Improve Echocardiograph Diagnosis", *Technical Report TR 84-5*, Comp. Sci., U. of Alberta, Edmonton, Alberta, July 1984.
- [10] W.A. Davis and Y.W. Tam, "An Efficient Method for Displaying 3D Images", *Conf. Proc. - Mini and Microcomputers and Their Application*, June 1985, pp. 201-204.
- [11] J.D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*,

Addison-Wesley, Reading, Massachusetts, 1982.

- [12] K.S. Fu and J.K. Mui, "A Survey on Image Segmentation", *Pattern Recognition*, Vol. 13, 1981, pp. 3-16.
- [13] H. Fuchs, G.D. Abram, and E.D. Grant, "Near Real-Time Shaded Display of Rigid Objects", *Computer Graphics*, Vol. 17, July 1983, pp. 65-68.
- [14] H. Fuchs, Z.M. Kedem, and S.P. Uselton, "Optimal Surface Reconstruction from Planar Contours", *Comm. ACM*, Vol. 20, Oct. 1977, pp. 693-702.
- [15] C.J. Gibson, "A New Method for the Three-Dimensional Display of Tomographic Images", *Phys. Med. Biol.*, Vol. 28, No. 10, 1983, pp. 1153-1157.
- [16] J.E. Gillespie, J.E. Adams and I. Isherwood, "Three-dimensional computed tomographic reformationsof sellar and para-sellar lesions", *Neuroradiology*, Vol. 29, 1987, pp. 30-35.
- [17] R. Gillespie and W.A. Davis, "Tree Data Structures for Graphics and Image Processing", *Proc. of the 7th Conf. of CMCCS*, June 1981, pp. 155-161.
- [18] D. Gordon and R.A. Reynolds, "Image Space Shading of 3-Dimensional Objects", *Computer Vision, Graphics, and Image Processing*, Vol. 29, 1985, pp. 361-376.
- [19] H. Gouraud, "Computer Display of Curved Surfaces", *IEEE Trans. Computers*, Vol. C-20, June 1971, pp. 623-629.
- [20] C.N. de Graaf et al., "Pyramid Segmentation of Medical 3D Images", *IEEE Proc. of the 1984 International Joint Alpine Symposium*, Feb. 1984, pp. 71-77.
- [21] P.B. Heffernan and R.A. Robb, "A New Method for Shaded Surface Display of Biological and Medical Images", *IEEE Transaction on Medical Imaging*, Vol. MI-4, No. 1, March 1985, pp. 26-38.
- [22] G.T. Herman, "Applications of Three-Dimensional Computer Graphics to Surgical Planning and Evaluation", *Computer Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp. 67-75.
- [23] G.T. Herman, "Three-Dimensional Computer Graphics Display in Medicine", *Computer Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp.

131-147.

- [24] G.T. Herman and H.K. Liu, "Three-Dimensional Display of Human Organs from Computer Tomograms", *Computer Graphics and Image Processing*, Vol. 9, 1979, pp. 1-21.
- [25] G.T. Herman and J.K. Udupa, "Display of Three-Dimensional Discrete Surfaces", *Proc. SPIE*, Vol. 283, 1981, pp. 90-97.
- [26] E.C. Hildreth, "Edge Detection", *MIT Artificial Intelligence Lab*, AI Memo No. 858, September 1985.
- [27] K.H. Hohne, R. Bernstein, "Shading 3D-Images from CT Using Gray-Level Gradients", *IEEE Transactions on Medical Imaging*, Vol. MI-5, No. 1, March 1986, pp. 45-47.
- [28] E. Keppel, "Approximating Complex Surfaces by Triangulation of Contour Lines", *IBM J. Res. Development*, Vol. 19, Jan. 1975, pp. 1-96.
- [29] H.K. Liu, "Two- and Three-Dimensional Boundary Detection", *Computer Graphics and Image Processing*, Vol. 6, 1977, pp. 123-134.
- [30] S.M. Lai, X. Li and W.F. Bischof, "Automated Detection of Breast Tumor", *Vision Interface '88*, June 1988, pp. 35-40.
- [31] D.L. McShan, S. Moffitt, and A.S. Glicksman, "Computer Graphics for Radiotherapy Treatment Planning", *Computer Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp. 89-95.
- [32] D.J. Meagher, "Geometric Modeling Using Octree-Encoding", *Computer Graphics and Image Processing*, Vol. 19, 1982, pp. 129-147.
- [33] D.J. Meagher, "Interactive Solids Processing for Medical Analyses and Planning", *Computer Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp. 97-107.
- [34] T. Peli and D. Malah, "A Study of Edge Detection Algorithms", *Computer Graphics and Image Processing*, Vol. 20, 1982, pp. 1-21.
- [35] B.T. Phong, "Illumination for Computer Generated Pictures", *Comm. ACM*, Vol. 18,

June 1987, pp. 311-317.

- [36] A. Rosenfeld and A.C. Kak, *Digital Signal Processing*, Vol. 1 and 2. 2nd Edition, 1982.
- [37] P.K. Sahoo et. al., "A Survey of Thresholding Techniques", *Computer Vision, Graphics and Image Processing*, Vol. 41, No. 2, February 1988, pp. 233-260.
- [38] H. Scharnweber, K.D. Tonnies, "Three-Dimensional Reconstruction and Display of Complex Anatomical Objects", *IEEE Proc. of the 1984 International Joint Alpine Symposium*, Fen. 1984, pp. 7-11.
- [39] G.B. Shaw, "Local and Regional Edge Detectors: Some Comparisons", *Computer Graphics and Image Processing*, Vol. 9, 1979, pp. 135-149.
- [40] W. Tiller, "Rational B-Splines for Curve and Surface Representation", *IEEE Computer Graphics and Applications*, Sept. 1983, pp. 61-69.
- [41] J.K. Udupa, "Determination of 3-D Shape Parameters from Boundary Information", *Computer Graphics and Image Processing*, Vol. 17, 1981, pp. 52-59.
- [42] J.K. Udupa, "Interactive Segmentation and Boundary Surface Formation for 3-D Digital Images", *Computer Graphics and Image Processing*, Vol. 18, 1982, pp. 213-235.
- [43] M.W. Vannier, J.L. Marsh, and J.D. Warren, "Three-Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation", *Computer Graphics*, Vol. 17, July 1983, pp. 263-266.
- [44] M.W. Vannier, J.L. Marsh, W.G. Stevens, and D.M. Dye, "Surface Reconstruction and Computer-Aided Design of Craniofacial Surgical Procedures Based on CT Scans", *Computers Graphics '84 Proc. of the 5th Annual Conf. and Exposition*, Vol. 1, 1984, pp. 117-129.
- [45] J.G. Verly, "High-Resolution Imaging in 3-D Reconstructive Tomography", *Computer Graphics and Image Processing*, Vol. 16, 1981, pp. 1-19.
- [46] J.S. Weszka, "A Survey of Threshold Selection Techniques", *Computer Graphics and Image Processing*, Vol. 7, 1978, pp. 259-265.
- [47] B. Wordenweber, "Surface Triangulation for Picture Production", *IEEE Computer*

Graphics and Applications, Nov. 9183, pp. 45-51.

- [48] S.H. Wu, J.F. Abel, and D.P. Greenberg, "An Interactive Computer Graphics Approach to Surface Representation", *Comm. ACM*, Vol. 20, Oct. 1977, pp. 703-712.